

501 P1375 US00 #5  
5-10-02  
9M

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

JC821 U.S. PRO  
09/944749  
08/31/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年 9月 1日

出 願 番 号

Application Number:

特願2000-266100

出 願 人

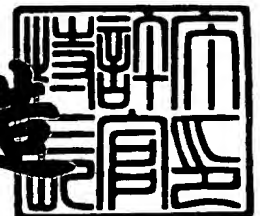
Applicant(s):

ソニー株式会社

2001年 6月 1日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造



出証番号 出証特2001-3051643

【書類名】 特許願

【整理番号】 0000494703

【提出日】 平成12年 9月 1日

【あて先】 特許庁長官 及川 耕造 殿

【国際特許分類】 G11B 20/12

【発明者】

【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社  
内

【氏名】 石坂 敏弥

【特許出願人】

【識別番号】 000002185

【氏名又は名称】 ソニー株式会社

【代表者】 出井 伸之

【代理人】

【識別番号】 100082762

【弁理士】

【氏名又は名称】 杉浦 正知

【電話番号】 03-3980-0339

【手数料の表示】

【予納台帳番号】 043812

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9708843

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ改竄チェック装置および方法、ならびに、記録媒体

【特許請求の範囲】

【請求項 1】 記録媒体に記録されたデータが改竄されていないかどうかチェックするデータ改竄チェック装置において、

1 または複数のファイルをまとめた上位概念であるディレクトリに属する 1 または複数のファイルのそれぞれについて、ファイルの属性情報に基づき一意且つ逆演算による可逆性を持たない所定の演算方法で上記ファイル毎に演算された複数の第 1 の演算値を含むリスト型データ構造を、1 または複数含むデータブロックが、記録媒体上のファイルシステムがアクセスしない領域に記録されると共に、上記第 1 の演算値が対応するファイルにそれぞれ書き込まれた記録媒体から、上記データブロックと上記ファイルとをそれぞれ読み出す読み出し手段と、

上記読み出し手段によって読み出された上記ファイルの属性情報に基づき上記所定の演算方法で演算された第 2 の演算値と、上記読み出し手段によって読み出された上記データブロックのうち、上記読み出し手段によって読み出された該ファイルが属する上記ディレクトリに対応した上記リスト型データ構造に含まれる、該ファイルに対応した上記第 1 の演算値とを比較する比較手段とを有し、

上記比較手段による比較結果に基づき、上記第 2 の演算値と上記第 1 の演算値とが一致しないときに、該ファイルが改竄されたと判断することを特徴とするデータ改竄チェック装置。

【請求項 2】 請求項 1 に記載のデータ改竄チェック装置において、

上記記録媒体に記録される上記リスト型データ構造には、該リスト型データ構造に含まれる上記ディレクトリのそれぞれについて、該ディレクトリに属する全ての上記ファイルの上記第 1 の演算値を用いて上記所定の演算方法によって演算された第 3 の演算値がさらに含まれると共に、上記記録媒体に記録される上記データブロックには、該データブロックに含まれる全ての上記リスト型データ構造の上記第 3 の演算値を用いて上記所定の演算方法で演算された第 4 の演算値がさらに含まれることを特徴とするデータ改竄チェック装置。

【請求項 3】 請求項 2 に記載のデータ改竄チェック装置において、

上記読み出し手段による上記記録媒体からの上記ファイルの読み出し時に、対象となる上記ディレクトリに対応する、上記読み出し手段によって読み出された上記データブロックに含まれる上記リスト型データ構造の上記第 3 の演算値と、上記対象となる上記ディレクトリに属する全ての上記ファイルの上記第 1 の演算値を用いて上記所定の演算方法で演算された第 5 の演算値とを比較し、比較結果に基づき上記第 3 および第 5 の演算値が一致しなければ、上記ディレクトリが改竄されたと判断することを特徴とするデータ改竄チェック装置。

【請求項 4】 請求項 1 に記載のデータ改竄チェック装置において、

上記ファイルは、該ファイルの種類に応じて上記ディレクトリにまとめられることを特徴とするデータ改竄チェック装置。

【請求項 5】 請求項 4 に記載のデータ改竄チェック装置において、

上記ファイルの種類がデータ改竄チェックを必要とされていないファイルを示すときには、該ファイルの種類が対応する上記ディレクトリに対する上記データ改竄チェックを行わないようにしたことを特徴とするデータ改竄チェック装置。

【請求項 6】 請求項 4 に記載のデータ改竄チェック装置において、

上記ファイルの種類は、該ファイルに格納されるデータ内容に基づくことを特徴とするデータ改竄チェック装置。

【請求項 7】 請求項 4 に記載のデータ改竄チェック装置において、

上記ファイルの種類は、該ファイルに対する保護の有無に基づくことを特徴とするデータ改竄チェック装置。

【請求項 8】 請求項 1 に記載のデータ改竄チェック装置において、

記録媒体にデータを記録する記録手段をさらに有し、

上記データブロックは、該データブロックの更新回数または該データブロックが無効であることを示す値が格納されるフィールドを有し、上記記録手段は、上記データブロックをファイルシステムがアクセスしない領域に 2 重書きし、該データブロックが書き替えられる際に、該 2 重書きされた一方のデータブロックの上記フィールドの値を上記データブロックが無効であることを示す値にして該一方のデータブロックを書き替え、該書き替えが終了したら、該一方のデータブロ

ックの上記フィールドの値を該一方のデータブロックの更新回数を示す値にする  
と共に、上記2重書きされた他方のデータブロックの上記フィールドの値を上記  
データブロックが無効であることを示す値にして該他方のデータブロックを書き  
替え、該書き替えが終了したら、該他方のデータブロックの上記フィールドの値  
を該他方のデータブロックの更新回数を示す値とするようにしたことを特徴とす  
るデータ改竄チェック装置。

【請求項9】 請求項1に記載のデータ改竄チェック装置において、

上記第1の演算値は、対応するファイルがファイルシステムで構築される木構  
造の探索順に並べられて上記リスト型データ構造に格納されることを特徴とする  
データ改竄チェック装置。

【請求項10】 記録媒体に記録されたデータが改竄されていないかどうか  
チェックするデータ改竄チェック方法において、

1または複数のファイルをまとめた上位概念であるディレクトリに属する1ま  
たは複数のファイルのそれぞれについて、ファイルの属性情報に基づき一意且つ  
逆演算による可逆性を持たない所定の演算方法で上記ファイル毎に演算された複  
数の第1の演算値を含むリスト型データ構造を、1または複数含むデータブロッ  
クが、記録媒体上のファイルシステムがアクセスしない領域に記録されると共に  
、上記第1の演算値が対応するファイルにそれぞれ書き込まれた記録媒体から、  
上記データブロックと上記ファイルとをそれぞれ読み出す読み出しのステップと

上記読み出しのステップによって読み出された上記ファイルの属性情報に基づ  
き上記所定の演算方法で演算された第2の演算値と、上記読み出しのステップに  
よって読み出された上記データブロックのうち、上記読み出しのステップによっ  
て読み出された該ファイルが属する上記ディレクトリに対応した上記リスト型デ  
ータ構造に含まれる、該ファイルに対応した上記第1の演算値とを比較する比較  
のステップと  
を有し、

上記比較のステップによる比較結果に基づき、上記第2の演算値と上記第1の  
演算値とが一致しないときに、該ファイルが改竄されたと判断することを特徴と

するデータ改竄チェック方法。

【請求項 1 1】 1 または複数のファイルをまとめた上位概念であるディレクトリを有するファイル構造でデータが記録される記録媒体において、

1 または複数のファイルをまとめた上位概念であるディレクトリに属する 1 または複数のファイルのそれぞれについて、ファイルの属性情報に基づき一意且つ逆演算による可逆性を持たない所定の演算方法で上記ファイル毎に演算された複数の第 1 の演算値を含むリスト型データ構造を、1 または複数含むデータブロックが、ファイルシステムがアクセスしない領域に記録されると共に、上記第 1 の演算値が対応するファイルにそれぞれ書き込まれることを特徴とする記録媒体。

【請求項 1 2】 請求項 1 1 に記載の記録媒体において、

上記リスト型データ構造には、該リスト型データ構造に含まれる上記ディレクトリのそれぞれについて該ディレクトリに属する全ての上記ファイルの上記第 1 の演算値を用いて上記所定の演算方法によって演算された第 2 の演算値がさらに含まれると共に、上記データブロックには、該データブロックに含まれる全ての上記リスト型データ構造の上記第 2 の演算値を用いて上記所定の演算方法で演算された第 3 の演算値がさらに含まれることを特徴とする記録媒体。

【請求項 1 3】 請求項 1 1 に記載の記録媒体において、

上記ファイルは、該ファイルの種類に応じて上記ディレクトリにまとめられることを特徴とする記録媒体。

【請求項 1 4】 請求項 1 3 に記載の記録媒体において、

上記ファイルの種類は、該ファイルに格納されるデータ内容に基づくことを特徴とするデータ改竄チェック装置。

【請求項 1 5】 請求項 1 3 に記載の記録媒体において、

上記ファイルの種類は、該ファイルに対する保護の有無に基づくことを特徴とする記録媒体。

【請求項 1 6】 請求項 1 1 に記載の記録媒体において、

上記データブロックは、ファイルシステムがアクセスしない領域に 2 重書きされると共に、該データブロックの更新回数または該データブロックが無効であることを示す値が格納されるフィールドを有し、該データブロックが書き替えられ

る際に、該 2 重書きされた一方のデータブロックの上記フィールドの値を上記データブロックが無効であることを示す値にして該一方のデータブロックを書き替え、該書き替えが終了したら、該一方のデータブロックの上記フィールドの値を該一方のデータブロックの更新回数を示す値にすると共に、上記 2 重書きされた他方のデータブロックの上記フィールドの値を上記データブロックが無効であることを示す値にして該他方のデータブロックを書き替え、該書き替えが終了したら、該他方のデータブロックの上記フィールドの値を該他方のデータブロックの更新回数を示す値とするようにしたことを特徴とする記録媒体。

【請求項 1 7】 請求項 1 1 に記載の記録媒体において、

上記データブロックのサイズは、該データブロックが記録される記録媒体上に存在する上記ディレクトリの数に応じて可変長とされることを特徴とする記録媒体。

【請求項 1 8】 請求項 1 1 に記載の記録媒体において、

上記リスト型データ構造は、該リスト型データ構造が対応する上記ディレクトリに属する上記ファイル数に応じて可変長とされることを特徴とする記録媒体。

【請求項 1 9】 請求項 1 1 に記載の記録媒体において、

上記第 1 の演算値は、対応するファイルがファイルシステムで構築される木構造の探索順に並べられて上記リスト型データ構造に格納されることを特徴とする記録媒体。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

この発明は、著作権保護され、記録媒体に記録されたコンテンツファイルに対する改竄の有無をチェックするようなデータ改竄チェック装置および方法、ならびに、記録媒体に関する。

【 0 0 0 2 】

【従来の技術】

近年、DVD (Digital Versatile Disk) に代表される、高密度光ディスクの開発が進み、それに伴い、規格の標準化が進められた。この標準化により、UDF

(Universal Disk Format)が策定された。DVDが書き換え可能とされたDVD-RAM(DVD-Random Access Memory)は、このUDFに従った論理フォーマットが用いられる。また、CD-ROM(Compact Disc-Read Only Memory)が書き込み可能とされたCD-Rや、書き換え可能とされCD-RWも、このUDFを適用可能である。

#### 【0003】

UDFにおいては、階層的なファイルシステムが用いられ、ルートディレクトリに格納された情報からサブディレクトリが参照され、サブディレクトリに格納された情報から、さらに別のサブディレクトリの参照や、実体的なファイルの参照がなされる。

#### 【0004】

上述について、より具体的に説明する。ディスク上の記録領域は、セクタを最小単位としてアクセスされ、例えばDVD-RAMでは、ディスクの内側から外側へとアクセスがなされる。最内周側から、リードイン領域に続けてボリューム情報が書き込まれる領域（ここでは、システム領域とする）が配され、ここに、ルートディレクトリのファイルエントリ(File Entry:以下、FEと略称する)が書き込まれる位置が示される。FEは、ルートディレクトリ、サブディレクトリおよびファイルのアドレスと長さの情報であるアロケーションディスクリプタ(Allocation Descriptor:以下、ADと略称する)からなる。

#### 【0005】

ルートディレクトリのFEにおいて、ADによって実体としてのルートディレクトリの論理アドレスと長さとが示される。ルートディレクトリは、1または複数のファイル識別記述子(File Identifier Descriptor:以下、FIDと略称する)を含み、FIDによって、ルートディレクトリ下にあるサブディレクトリのFEやファイルのFEが参照される。これらのFEによって、それぞれ対応するサブディレクトリやファイルの実体が参照される。また、サブディレクトリの実体は、さらに1または複数のFIDを含む。すなわち、UDFにおいて、ルートディレクトリ以外は、FIDおよびFEをポインタとして、FID、FEおよび実体の順にアクセスが行われる。



## 【 0 0 0 6 】

ところで、記録媒体に記録された情報データの著作権保護に関する技術要素の一つとして、改竄チェックがある。これは、記録された情報データに対する不正な改竄の防止と、情報データが不正に改竄されてしまった場合の改竄の検出を行うための技術である。さらに、この改竄チェックは、情報データを、その情報データが記録された記録媒体で閉じたものとして縛ることで、情報データの不正な複写を防止することができるので、著作権保護においては、非常に重要な技術要素であるということがいえる。

## 【 0 0 0 7 】

従来のデータ改竄チェック方法について、概略的に説明する。情報データの単位であるファイル（コンテンツ）毎に、当該ファイルの重要管理情報、著作権情報および状態情報などの属性情報から、一意に生成され、且つ、逆演算による源情報の同定が困難であるような改竄チェック値を求める。求められた改竄チェック値は、記録媒体上に設けられた、ユーザが容易にアクセスできないエリアやファイルである改竄チェック管理値スペースに書き込まれる。当該記録媒体を再生したときに、再生データに基づき改竄チェック値を求め、求められた改竄チェック値と、改竄チェック値管理スペースに書き込まれた改竄チェック値とを比較することで、不正なファイルの移動や複写が行われたかどうかをチェックする。

## 【 0 0 0 8 】

同様に、同一記録媒体上に存在する全てのファイルを対象に、改竄チェック値を求めることで、不正なファイルの移動や複写が行われたかどうかをチェックする方法も、実施されている。

## 【 0 0 0 9 】

改竄チェック値を求める方法としては、例えば I S O / I E C 9 7 9 7 に規定されている M A C (Message Authentication Code) と称される演算方法が知られている。ここで、ファイル毎の改竄チェック値そのものを指して M A C と呼ぶことがあり、このとき特に、同一記録媒体上にある全てのファイルが対象とされた改竄チェックを I C V (Integrity Check Value) と呼んで、両者の差別化を図ることがある。以下では、ファイル毎の改竄チェック値を M A C と称し、同一記録

媒体上の全てのファイルが対象にされた改竄チェック値を I C V と称する。

【 0 0 1 0 】

【発明が解決しようとする課題】

従来の改竄チェック方法では、上述したように、記録媒体上のファイル毎に M A C を生成する。一方、近年では、記録媒体の大容量化が著しく進んでいる。記録媒体の容量が大きくなればなるほど記録媒体上に存在するファイル数も膨大になり、それに伴い、M A C の量も膨大になることが予想される。上述した改竄チェック方法の手続として、チェックを行うタイミングにおいて、記録媒体上に存在するファイルなどにに基づき現状の M A C および I C V を演算することが必要となる。その際に、特にディスク状の記録媒体においては、その構造上、駆動系が存在し、M A C および I C V を求める情報を得るためになされるアクセスに要する時間が多くかかってしまうという問題点があった。また、これは、ユーザによってもストレスになるという問題点があった。

【 0 0 1 1 】

さらに、同一記録媒体上に、様々な種類のコンテンツが存在する場合、この記録媒体を再生するアプリケーションソフトウェアやセット機器によっては、対象外になるようなコンテンツファイルについても、毎回改竄チェック手続に含まれることになる。このような場合、改竄チェックに費やされる時間が余計にかかってしまうことになり、時間効率が悪いという問題点があった。

【 0 0 1 2 】

さらにまた、あるタイミングで特定のファイルについて改竄チェック手続を行わなければならない場合も考えられる。このような場合、アプリケーションソフトウェアやセット機器では、当該記録媒体についての最初の改竄チェックの際に保持された、当該ファイルの M A C リストの中から、対応する M A C を検索し、検索された M A C を用いて改竄チェックを行う。この場合、上述のように記録媒体が大容量となり、記録媒体上に存在するファイル数が膨大となると、M A C リストも膨大となり、そのリストの中から対応する M A C を特定するには多くの時間を費やす必要があり、上述と同様に、時間効率が悪いという問題点があった。

【 0 0 1 3 】

この問題に対する解決策の一例として、MACを格納するエリアを固定長として、ファイル番号などと合わせて直接的にMACの格納場所を特定する方法がある。しかしながら、大容量の記録媒体の場合、そのコンテンツ格納能力からかなりの大きさのエリアをそのために確保する必要があるという問題点があった。

【0014】

また、この方法では、MACを格納するエリアが固定長とされるために、ファイル数などを制限する必要がある、実用的とはいえないという問題点があった。

【0015】

したがって、この発明の目的は、データの改竄チェックを効率的に行うことができるデータ改竄チェック装置および方法、ならびに、記録媒体を提供することにある。

【0016】

【課題を解決するための手段】

この発明は、上述した課題を解決するために、記録媒体に記録されたデータが改竄されていないかどうかチェックするデータ改竄チェック装置において、1または複数のファイルをまとめた上位概念であるディレクトリに属する1または複数のファイルのそれぞれについて、ファイルの属性情報に基づき一意且つ逆演算による可逆性を持たない所定の演算方法でファイル毎に演算された複数の第1の演算値を含むリスト型データ構造を、1または複数含むデータブロックが、記録媒体上のファイルシステムがアクセスしない領域に記録されると共に、第1の演算値が対応するファイルにそれぞれ書き込まれた記録媒体から、データブロックとファイルとをそれぞれ読み出す読み出し手段と、読み出し手段によって読み出されたファイルの属性情報に基づき所定の演算方法で演算された第2の演算値と、読み出し手段によって読み出されたデータブロックのうち、読み出し手段によって読み出されたファイルが属するディレクトリに対応したリスト型データ構造に含まれる、ファイルに対応した第1の演算値とを比較する比較手段とを有し、比較手段による比較結果に基づき、第2の演算値と第1の演算値とが一致しないときに、ファイルが改竄されたと判断することを特徴とするデータ改竄チェック装置である。

## 【 0 0 1 7 】

また、この発明は、記録媒体に記録されたデータが改竄されていないかどうかチェックするデータ改竄チェック方法において、1または複数のファイルをまとめた上位概念であるディレクトリに属する1または複数のファイルのそれぞれについて、ファイルの属性情報に基づき一意且つ逆演算による可逆性を持たない所定の演算方法でファイル毎に演算された複数の第1の演算値を含むリスト型データ構造を、1または複数含むデータブロックが、記録媒体上のファイルシステムがアクセスしない領域に記録されると共に、第1の演算値が対応するファイルにそれぞれ書き込まれた記録媒体から、データブロックとファイルとをそれぞれ読み出す読み出しのステップと、読み出しのステップによって読み出されたファイルの属性情報に基づき所定の演算方法で演算された第2の演算値と、読み出しのステップによって読み出されたデータブロックのうち、読み出しのステップによって読み出されたファイルが属するディレクトリに対応したリスト型データ構造に含まれる、ファイルに対応した第1の演算値とを比較する比較のステップとを有し、比較のステップによる比較結果に基づき、第2の演算値と第1の演算値とが一致しないときに、ファイルが改竄されたと判断することを特徴とするデータ改竄チェック方法である。

## 【 0 0 1 8 】

また、この発明は、1または複数のファイルをまとめた上位概念であるディレクトリを有するファイル構造でデータが記録される記録媒体において、1または複数のファイルをまとめた上位概念であるディレクトリに属する1または複数のファイルのそれぞれについて、ファイルの属性情報に基づき一意且つ逆演算による可逆性を持たない所定の演算方法でファイル毎に演算された複数の第1の演算値を含むリスト型データ構造を、1または複数含むデータブロックが、ファイルシステムがアクセスしない領域に記録されると共に、第1の演算値が対応するファイルにそれぞれ書き込まれることを特徴とする記録媒体である。

## 【 0 0 1 9 】

上述したように、この発明によれば、1または複数のファイルをまとめた上位概念であるディレクトリに属する1または複数のファイルのそれぞれについて、

ファイルの属性情報に基づき一意且つ逆演算による可逆性を持たない所定の演算方法でファイル毎に演算された複数の第 1 の演算値を含むリスト型データ構造を、1 または複数含むデータブロックが、記録媒体上のファイルシステムがアクセスしない領域に記録されると共に、第 1 の演算値が対応するファイルにそれぞれ書き込まれた記録媒体から、データブロックとファイルとをそれぞれ読み出して、読み出されたファイルの属性情報に基づき所定の演算方法で演算された第 2 の演算値と、読み出されたデータブロックのうち、読み出し手段によって読み出されたファイルが属するディレクトリに対応したリスト型データ構造に含まれる、ファイルに対応した第 1 の演算値とを比較し、比較結果に基づき、第 2 の演算値と第 1 の演算値とが一致しないときに、ファイルが改竄されたと判断するようにしているため、時間的に効率よくデータ改竄チェックを行うことができる。

## 【 0 0 2 0 】

## 【発明の実施の形態】

以下、この発明の実施の一形態を、図面を参照しながら説明する。先ず、理解を容易とするために、記録媒体にディスクを用いた場合における基本的なデータ改竄チェック方法について、図 1 および図 2 を用いて説明する。ここで説明する基本的な改竄チェック方法は、ISO/IEC 9797 の MAC (Message Authentication Code) による演算方法を導入したものである。なお、以下では、ファイル毎に求められる改竄チェック値そのものを MAC 値と称し、同一記録媒体上にある全てのファイルを対象として求められる改竄チェック値を、ICV (Integrity Check Value) と称する。

## 【 0 0 2 1 】

図 1 は、ディスク記録媒体に新たにファイル # 1 および # 2 が追加されたときの改竄チェック方法を概略的に示す一例の機能ブロック図である。ディスク記録媒体（以下、ディスクと略称する）110 のユーザデータ領域 110 A に記録されたファイルのそれぞれについて、MAC の演算手法に基づき MAC 値が求められる。例えば、図 1 に示されるファイル # 1 のように、著作権情報やファイルの重要情報などのファイルの属性情報と、当該ファイルに固有な鍵となる情報、例えばそのファイルの実データ部分を暗号化する際に用いられたコンテンツ鍵とを

用い、MAC演算部 1 1 2 Aによって所定の演算がなされ、MAC値 # 1 が求められる。

#### 【 0 0 2 2 】

他のファイルについても同様にしてMAC値が求められる。例えばファイル # 2 では、上述と同様にコンテンツ鍵と、ファイルの重要情報および著作権保護条件などのファイルの属性情報から、MAC演算部 1 1 2 BによってMAC値 # 2 が求められる。

#### 【 0 0 2 3 】

MAC値は、上述の著作権情報、ファイルの重要情報およびコンテンツ鍵などに限らず、ファイルの再生回数や複製回数などをさらに用いて求めるようにしてもよい。例えば、ファイルの複製回数をさらに用いてMAC値を求めることにより、ファイルのコピー世代の制限を行うことができる。

#### 【 0 0 2 4 】

なお、MAC値は、与えられた値に基づき一方向性の関数であるハッシュ関数を用いて生成される値であり、MAC値から逆演算を行って元の値を求めることはできない。

#### 【 0 0 2 5 】

このようにして、ファイル # 1 および # 2 についてそれぞれ生成されたMAC値 # 1 および # 2 は、それぞれ対応するファイルのヘッダ情報として、対応するファイルに格納される。それと共に、MAC値 # 1 および # 2 は、ディスク 1 1 0 の記録領域のうち、ユーザが容易にアクセスできない領域、例えばディスク 1 1 0 に用意されたリードイン領域 1 1 0 Bにまとめて格納される。リードイン領域 1 1 0 Bは、そのディスク 1 1 0 が扱われるシステムにおいて、ファイルシステムがアクセスしない領域、すなわち、当該ファイルシステムにおいて論理アドレス上では存在しない領域である。

#### 【 0 0 2 6 】

さらに、ファイル毎に生成されたMAC値 # 1 および # 2 は、MAC演算部 1 1 3に入力される。MAC演算部 1 1 3には、さらにICV演算用鍵が入力され、このICV演算用鍵と、入力されたMAC値 # 1 および # 2 とからICVを生

成する。このICVは、ディスク110上において対象となる全てのファイルの情報が反映された値となっている。生成されたICVは、上述したMAC値#1および#2と同様に、ディスク110のリードイン領域110Bに記録される。

## 【0027】

このように、ディスク110上に新規にファイル#1および#2が追加された場合には、リードイン領域110Bには、ファイル毎の改竄チェック値であるMAC値#1および#2と、ディスク110上において対象となる全てのファイルの情報が反映された改竄チェック値であるICVとが記録される。ここでは、改竄チェックに関連する情報が格納される、リードイン領域110Bのような領域およびその領域におけるデータ構造を、シーケンスブロックと称する。図1において、シーケンスブロック114に対してファイル毎のMAC値#1、#2と、ICVとが格納される。シーケンスブロック114は、上述したように、ディスク110の記録領域のうち、ユーザが容易にアクセスできない、例えばリードイン領域110Bに記録される。

## 【0028】

図2は、ディスク110上で、例えばある特定のファイルを再生／移動する場合や、システムにおいて改竄チェックが要求されたタイミングにおける一例の手続を示す機能ブロック図である。ある特定のファイルについての改竄チェックは、MAC値を用いて行われる。例えばファイル#1が再生あるいは移動される場合、先ず、図1を用いて既に説明した方法により、当該ファイルに関して、MAC値が生成される。ファイル#1に関して生成されるこのMAC値を、MAC値#1'とする。

## 【0029】

一方、ファイル#1がこのディスク110に記録された際に生成され、ファイル#1のヘッダ部に格納されたMAC値#1と、ヘッダ部へのMAC#1の格納と共にリードイン領域110Bのシーケンスブロック114に格納されたMAC#1とが取り出される。取り出されたこれらのMAC値#1と、上述のMAC値#1'とが比較部115Aにおいて比較される。この比較により、MAC値#1'がヘッダ部から取り出されたMAC値#1およびシーケンスブロック114か

ら取り出されたMAC値#1と一致しない場合には、ファイル#1に対して不正に改竄が行われたおそれがあると判断され、システムにおいてエラー処理がなされる。

#### 【0030】

また、ディスク110上の全てのファイルを対象とした改竄チェックは、シーケンスブロック114に格納されたICVを用いて行われる。例えばディスク110上のあるファイル（ファイル#1とする）が再生あるいは移動される場合、先ず、上述と同様にして、ディスク110上に存在する全てのファイルについてMAC値が生成され、生成された、全てのファイルについてもMAC値に対して、MAC演算部113において、ICV演算用鍵を用いてICV'が生成される。

#### 【0031】

一方、ディスク110のリードイン領域110Bのシーケンスブロック114に格納されたICVが取り出される。シーケンスブロック114から取り出されたこのICVと、ディスク110上の全てのファイルのMAC値から生成されたICV'とが比較部16で比較される。比較の結果、両者が一致しなければ、ディスク110上においてファイルの不正な移動やコピー、消去などが行われたと判断され、システムにおいてエラー処理がなされる。

#### 【0032】

上述の基本的な改竄チェック方法を踏まえて、この発明の実施の第1の形態による改竄チェック方法について説明する。図3は、この実施の一形態による改竄チェック値のデータ構造を概略的に示す。この実施の第1の形態は、上述の図1および図2で示した基本構成における、シーケンスブロック114の構造にポイントを置いたものである。適用される記録媒体は、上述したような、ディスク状記録媒体であるディスク110が想定されている。なお、この実施の第1の形態では、適用可能な記録媒体はディスク状記録媒体に限られない。

#### 【0033】

シーケンスブロック114は、1または複数のシーケンスページ121A、121B、・・・とディスク110上の全てのファイルについての改竄チェック値



ICVとからなる。シーケンスページ121A、121B、・・・のそれぞれは、ディスク110上のファイル毎の改竄チェック値、例えばMAC値が複数、リスト構造として格納されるデータ構造とされている。各ファイルとMAC値との対応付けは、例えば、ファイルシステムによって構築されるファイルの木構造の探索順に対応した順序で、シーケンスページにおけるMAC値のエントリを並べることとされる。

## 【0034】

また、ディスク110上のファイルは、1または複数のファイルが上位概念であるディレクトリとしてまとめられる。シーケンスページ#1、#2、・・・のそれぞれは、格納されるMAC値に対応するファイルがまとめられたディレクトリに対応付けられる。例えば、各ディレクトリ122A、122B、・・・と各シーケンスページ121A、121B、・・・とが互いに対応がとれるようなIDを格納するフィールドを、シーケンスページ121A、121B、・・・にそれぞれ規定することで、各ディレクトリ122A、122B、・・・と各シーケンスページ121A、121B、・・・との対応をとることができる。

## 【0035】

例えばシーケンスページ121A、121B、・・・は、ディスク110上に形成されたディレクトリ122A、122B、・・・とそれぞれ対応付けられている。シーケンスページ121A、121B、・・・は、対応するディレクトリに含まれるファイルそれぞれから生成されたMAC値と、当該ディレクトリに含まれる全てのファイルのMAC値を入力として生成されたICVとが格納される。

## 【0036】

一例として、ディレクトリ122Aと対応付けられるシーケンスページ121Aは、ディレクトリ122Aに格納されるファイル#1[0]、#1[1]、・・・、#1[n-1]（図示しない）のそれぞれから生成されたMAC値#1[0]、#1[1]、・・・、#1[n-1]と、ファイル#1[0]、#1[1]、・・・、#1[n-1]の全てのMAC値から生成されたICVとからなる。

## 【0037】

なお、以下では、ディレクトリに含まれる全てのファイルのMAC値を入力として生成されたICVを、D-ICVと称する。また、図3において、「#1」、「#2」は、対応するディレクトリを区別するため、括弧[]の内容は、当該ディレクトリに格納されるファイルを区別するために記される。

## 【0038】

さらに、上述したシーケンスブロック114に格納されるICVは、各シーケンスページ121A、121B、・・・に格納される全てのD-ICV#1、#2、・・・を入力として生成される。したがって、シーケンスブロック114に格納されるICVは、結果的に、ディスク110全体のファイルに対するMAC値が反映された値となる。

## 【0039】

上述したように、各シーケンスページ121A、121B、・・・に格納されるMAC値の数は、対応するディレクトリ122A、122B、・・・に格納されるファイル数に応じて変わる。同様に、シーケンスブロック114に格納されるシーケンスページ121A、121B、・・・の数は、ディスク110に作成されるディレクトリ122A、122B、・・・の数に応じて変わる。したがって、シーケンスブロック114は、可変長領域として構築されると共に、シーケンスブロック114中の各シーケンスページ121A、121B、・・・も、それぞれ可変長領域として構築される。

## 【0040】

図4は、ディスク110の一例の物理フォーマットを部分的に示す。ディスク110がディスク内周側から外周側へとデータを記録していく場合、図4の上側がディスク内周側、下側が外周側にそれぞれ相当する。ディスク内周側からリードイン領域110Bが配置されている。リードイン領域110Bの外側から、ユーザデータ領域が開始される。ユーザデータ領域は、論理アドレスが与えられ、このディスク110を扱うファイルシステムによってアクセス可能とされた領域である。一方、リードイン領域110Bは、上述したように、論理アドレス上では存在せず、ファイルシステムがアクセスしない領域である。

## 【 0 0 4 1 】

リードイン領域 1 1 0 B の先頭に、メディア I D が記録される領域 1 3 0 が例えば 1 3 2 バイトの大きさで設けられる。メディア I D の領域の後ろにエンボス領域 1 3 1 が配され、エンボス領域 1 3 1 の次から、リードイン領域 1 1 0 B の最後までが書き替え可能な領域 1 3 2 とされる。シーケンスブロック 1 1 4 は、この書き替え可能な領域 1 3 2 の所定位置に 2 重書きされる。シーケンスブロック 1 1 4 が 2 重書きされたそれぞれを、シーケンスブロック 1 1 4 A、シーケンスブロック 1 1 4 B とする。

## 【 0 0 4 2 】

この書き替え可能な領域 1 3 2 の所定位置から開始される、大きさが例えば 1 2 8 k バイトの固定領域 1 3 3 に、シーケンスブロック 1 1 4 A が書き込まれる。そして、固定領域 1 3 3 に続けて配される大きさが例えば 1 2 8 k バイト固定領域 1 3 3' には、シーケンスブロック 1 1 4 A と同一のデータからなるシーケンスブロック 1 1 4 B が書き込まれる。

## 【 0 0 4 3 】

例えば固定領域 1 3 3 において、シーケンスブロック 1 1 4 A は例えば前詰めで書き込まれ、固定領域 1 3 3 内で余った領域 1 3 4 は、スタッフィングバイトで埋められる。固定領域 1 3 3' も、固定領域 1 3 3 と同様な構成とされる。

## 【 0 0 4 4 】

図 5 は、シーケンスブロック 1 1 4 A および 1 1 4 B の一例の論理フォーマットを示す。図中、「0 x」から始まる数値は、1 6 進表記されていることを示す。ここでは、シーケンスブロック 1 1 4 A を例にとって説明する。横方向にバイトが示されている。最初の 2 バイトのフィールド「S P E N u m」には、このシーケンスブロック 1 1 4 A にエントリされているシーケンスページの総エントリ数が格納される。次の 4 バイトのフィールド「B l o c k S i z e」は、シーケンスブロック 1 1 4 A 自身のサイズが格納される。シーケンスブロック 1 1 4 A の先頭バイトから最終エントリの最終バイトまでのバイト数が格納される。

## 【 0 0 4 5 】

フィールド「B l o c k S i z e」に続く 4 バイトには、フィールド「R e

vision」が配される。フィールド「Revision」は、このシーケンスブロック114Aの書き替え回数およびこのシーケンスブロック114Aが有効／無効のどちらの状態にあるかを示す値Revisionが格納される。値Revisionは、初期状態が「0」とされ、このシーケンスブロック114Aが書き替えられる度に、値が「1」ずつ増加される。また、このシーケンスブロック114Aが無効であるか、または、書き替え途中である場合には、値Revision=Invalid Numとされ、その旨が示される。値Invalid Numは、例えば値「0xFFFFFFFF」で示される。

## 【0046】

「Revision」に続く6バイトおよびその次の16バイトは、システム予約されている。このシステム予約バイトにICVを格納することができる。133バイト目からシーケンスページ121A、121B、・・・がエントリされる。ディスク110上に存在する全てのディレクトリに対応したシーケンスページの情報がエントリされたら、例えば全て「0」からなるスタフピングバイトで以て、シーケンスブロック114Aが格納される固定領域133が、最終バイトまで埋め尽くされる。なお、フィールド「Block Size」にシーケンスブロック114A自身のサイズ情報が記述されているため、スタフピングバイトは、必ずしも必要ではない。

## 【0047】

このように、この実施の一形態では、シーケンスブロックが、ディスク110上に存在するディレクトリ毎に一対一に対応したデータ構造であるシーケンスページの集合体で構成される。

## 【0048】

図6は、シーケンスページ121A、121B、・・・の一例の論理フォーマットを示す。図中、「0x」から始まる数値は、16進表記されていることを示す。ここでは、シーケンスページ121Aを例にとって説明する。最初の2バイトのフィールド「Page ID」は、このシーケンスページ121Aとディスク110上のファイルシステムにおけるディレクトリとを関連付けるためのPage IDが格納される。次の2バイトのフィールド「Entry Num」は

、このシーケンスページ 1 2 1 A にエントリされている MAC 値の総数が格納される。次の 4 バイトのフィールド「P a g e   S i z e」は、シーケンスページ 1 2 1 A 自身のサイズが格納される。シーケンスページ 1 2 1 A の先頭バイトから最終エントリの最終バイトまでのバイト数が格納される。次の 8 バイトは、システム予約されている。

## 【 0 0 4 9 】

1 7 バイト目のフィールド「D - I C V」は、D - I C V が格納される。そして、続く 3 3 バイト目から MAC 値が格納される。MAC 値は、例えば 6 4 ビットすなわち 8 バイトの固定長とされている。MAC 値は、シーケンスページ 1 2 1 A の 1 7 バイト目から 8 バイトずつ詰め込まれる。

## 【 0 0 5 0 】

このように、この実施の一形態では、シーケンスページには、ディスク 1 1 0 上のディレクトリと一対一で対応をとれるような値が格納される、フィールド「P a g e   I D」が規定されている。そのため、改竄チェック手続においては、シーケンスブロック 1 1 4 A 内の特定のシーケンスページだけを対象とした処理を行うことができる。すなわち、この実施の第 1 の形態によれば、ディスク 1 1 0 上の特定のディレクトリだけを対象としてデータ改竄チェックを行うことができる。

## 【 0 0 5 1 】

この実施の一形態では、上述のシーケンスブロック 1 1 4 A および 1 1 4 B は、互いにバックアップとして機能する。すなわち、ディスク上のファイルの再生や変更、複写などがなされる場合に、一方は当該処理が開始されると共に更新され、他方は当該処理中は更新を禁止し、当該処理が終了すると共に更新されるようにする。これにより、当該処理途中に例えば電源が切られるなどしてシステムダウンしても、シーケンスブロック 1 1 4 A および 1 1 4 B のうち何方か一方は正常に残っており、ディスク 1 1 0 の状態を後に復帰させることが容易である。

## 【 0 0 5 2 】

図 7 は、この実施の一形態によるシーケンスブロック 1 1 4 A および 1 1 4 B の更新手続を示す。図 7 A および図 7 B は、シーケンスブロック 1 1 4 A およ

び114Bにおけるフィールド「Revision」の状態の遷移をそれぞれ示す。なお、図7において、図の上側から下側へ時間が進行しているものとする。時刻Aで、ディスク110上に記録されるファイルの更新や複写、あるいはファイルの新規の記録などのファイル処理が開始され、時刻Bで当該処理が終了するものとする。また、図7Cは、各時刻におけるフィールド「Revision」の値を示す。シーケンスブロック114Aおよび114Bのフィールド「Revision」の値は、当初、共に $Revision = n - 1$ であるとする。

## 【0053】

時刻Aでディスク110上のファイルに対するファイル処理が開始されると、シーケンスブロック114Bが一旦削除される。このとき、シーケンスブロック114Bのフィールド「Revision」だけが値を $Revision = Invalid\ Num$ に変更されて残される。あるいは、時刻Aでのファイル処理の開始と共に、既存のシーケンスブロック114Bが、値 $Revision = Invalid$ とされたフィールド「Revision」（およびフィールド「SPE Num」、フィールド「ブロックサイズ」）のみからなるシーケンスブロック114Bで上書きされる。

## 【0054】

一方、シーケンスブロック114Aは、時刻Aでディスク110上でのファイル処理が開始されても、何も変化されず、フィールド「Revision」の値も、 $Revision = n - 1$ のままとされる。

## 【0055】

時刻Bになり当該ファイル処理が終了されると、ファイル処理の結果を反映し、フィールド「Revision」の値が $Revision = n$ とされたシーケンスブロック114Bが書き込まれる。すなわち、フィールド「Revision」の値がファイル処理前の値よりも1だけ大きくされる。勿論、ディスク110上の全てのファイルおよびディレクトリに関するMAC値およびD-ICVも計算し直されシーケンスページが再構築され、シーケンスブロック114Bに格納される。

## 【0056】

このシーケンスブロック 114 B の書き込みが終了すると、シーケンスブロック 114 A の書き込みが開始される。先ず、シーケンスブロック 114 A のフィールド「Revision」の値が Revision=Invalid Num とされ、シーケンスブロック 114 A が無効化される（時刻 C）。そして、当該ファイル処理の結果を反映し、フィールド「Revision」の値が Revision=n とされたシーケンスブロック 114 A が書き込まれる。これ以降は、シーケンスブロック 114 A および 114 B は、同一データからなり、フィールド「Revision」においては同一の Revision=n を有する。

## 【0057】

このようにして、シーケンスブロック 114 A および 114 B の更新を行うことで、図 7 で示されるどの時刻をとってもシーケンスブロック 114 A および 114 B のどちらか一方は、必ず有効とされている。そのため、データの記録や複写などといったファイル処理の途中に、何らかの理由によりシステムがダウンしても、救済処理が可能である。

## 【0058】

すなわち、システム復帰した際に、現状におけるディスク 110 上のファイルおよびディレクトリから MAC 値、ICV および D-ICV を生成し、生成されたこれらの値と、シーケンスブロック 114 A および 114 B のうちシステム復帰直後に有効となっているシーケンスブロックの内容とを比較する。両者の差分を求めることで、何らかの救済措置をとることが可能である。

## 【0059】

図 8 は、この救済措置を含めたシーケンスブロックの更新手続きの一例の処理を示すフローチャートである。このフローチャートによる処理は、シーケンスブロック 114 A および 114 B に記録されているデータが更新される場合に実行されるものである。

## 【0060】

先ず、最初のステップ S10 で、シーケンスブロック 114 A および 114 B のフィールド「Revision」の値（それぞれ Revision #1、Revision #2 とする）が比較される。比較の結果、Revision #1 と

Revision # 2 とが一致していなければ、処理はステップ S 1 1 に移行する。

【0061】

ステップ S 1 1 では、シーケンスブロック 1 1 4 A および 1 1 4 B のうち、フィールド「Revision」の値が Revision ≠ Invalid Num であるシーケンスブロックのデータが読み出される。読み出される対象となるデータは、そのシーケンスブロックに格納された各シーケンスページのデータである。このデータをデータ A とする。

【0062】

次のステップ S 1 2 では、現状のディスク 1 1 0 上のデータに基づき MAC 演算がなされ、新たにシーケンスブロックのデータが生成される。このときには、上述のステップ S 1 1 と同様に、生成される対象となるデータは、ディスク 1 1 0 上のファイルやディレクトリに基づくシーケンスページである。このデータをデータ B とする。

【0063】

ステップ S 1 3 で、上述したデータ A とデータ B とが比較される。比較の結果、若し、データ A とデータ B とが一致していると判断されれば、何も問題が無かったとされ、このフローチャートによる一連の処理が終了され、通常処理に移行する。一方、ステップ S 1 3 で、データ A とデータ B とが一致していないと判断されれば、処理はステップ S 1 4 に移行する。

【0064】

ステップ S 1 4 では、データ A とデータ B との差分が抽出され、その差分に基づきディスク 1 1 0 上のファイルの救済措置がとられる。例えば、データ A とデータ B との差分を求めることで、ディスク 1 1 0 上のシーケンスブロックによれば存在しているはずだが実際には存在していないファイルやディレクトリを見つけ出すことができる。同様に、データ A とデータ B とを比較することで、所定のファイルの著作権情報などが書き替えられているために、シーケンスブロックに記録されている MAC 値とは異なる MAC 値となっているファイルなどを見つけ出すことができる。



## 【0065】

このようなファイルに対して、可能であれば救済措置を施す。例えば、データの所定位置にE O F (End OF File)を付加し、強制的にファイルを構成することができる。また、不正に複写や著作権情報の書き替えなどがなされたおそれがあるファイルについては、そのファイルに再生不可の属性を与えたり、そのファイルを削除するようにしてもよい。

## 【0066】

こうしてデータAおよびデータBの差分が抽出され、ファイル救済措置がなされた後に、図7を用いて上述したようにして、シーケンスブロック114Aおよび114Bの更新手続きがなされる。

## 【0067】

一方、上述したステップS10で、シーケンスブロック114Aおよび114Bのフィールド「Revision」の値であるRevision#1およびRevision#2が比較され、両者が一致していると判断されれば、処理はステップS15に移行する。ステップS15では、さらに、シーケンスブロック114Aおよび114Bについて、他のフィールドのデータが互いに一致しているかどうか判断される。若し、両者が一致していると判断されれば、何も問題が無かったとされ、このフローチャートによる一連の処理が終了され、通常処理に移行する。

## 【0068】

一方、ステップS15において、他のフィールドのデータが互いに一致していないと判断されれば、処理はステップS16に移行し、シーケンスブロック114Aあるいは20'のエラー訂正処理がなされる。シーケンスブロックのエラー訂正処理がなされたら、このフローチャートによる一連の処理が終了され、通常処理に移行する。

## 【0069】

ステップS16によるエラー訂正処理は、例えば次のようにして行うことができる。ディスク110上に存在するファイルやディレクトリ情報を取得し、取得されたそれらの情報に基づきファイル毎のMAC値やディレクトリ毎のD-I C

V、さらにはディスク 1 1 0 全体の I C V を生成し、ディスク 1 1 0 上の現状でのシーケンスブロックを作成する。作成されたこのシーケンスブロックと、ディスク 1 1 0 上から読み出されたシーケンスブロック 1 1 4 A および 1 1 4 B とをそれぞれ比較する。そして、シーケンスブロック 1 1 4 A および 1 1 4 B のうち、新たに生成されたシーケンスブロックと一致する方を基準にして、図 7 で上述したシーケンスブロックの更新手続きを行う。

#### 【 0 0 7 0 】

次に、この発明によるデータ改竄チェック方法の第 1 の応用例について説明する。第 1 の応用例は、ディスク 1 1 0 においてディレクトリ毎に同一種類のファイルをまとめて格納する場合に用いて好適な例である。より具体的には、1 枚のディスク 1 1 0 に対して作成される複数のディレクトリ A、B、C、・・・は、ディレクトリ A、B、C、・・・毎に異なる種類のファイルから構成される。例えば、ディレクトリ A は静止画像ファイル、ディレクトリ B は動画ファイル、ディレクトリ C は音声ファイルから、それぞれ構成される。このように、記録媒体として多用途のものを考えた場合、同一記録媒体上に多種のデータが存在することになる。

#### 【 0 0 7 1 】

また、改竄チェックに関しては、一般的に、記録媒体がアプリケーションにセットされたときや、ディスクドライブにディスク 1 1 0 が装填された状態で装置の電源を投入したときなど、比較的頻繁に行われることが予想される。同様に、記録媒体に対して新規に記録を行ったり、他の記録媒体との間でファイルの複写や移動が発生した場合には、記録媒体において I C V の再計算とシーケンスブロック 1 1 4 A および 1 1 4 B の更新が行われることになる。

#### 【 0 0 7 2 】

なお、ここでアプリケーションとは、コンピュータ装置に搭載されたアプリケーションソフトウェアや、ハードウェアとソフトウェアとが一体的に構成された専用機的な装置など、この記録媒体に対して記録／再生などを行える構成を指す。

#### 【 0 0 7 3 】

一方、アプリケーションによっては、記録媒体上に存在する全てのデータを、必ずしも扱えるとは限らない。例えば、音楽再生専用機の場合には、音楽ファイルから構成されるディレクトリのみを対象にし、そのディレクトリで閉じて改竄チェック手続を行えるようにすれば、時間的な効率を向上させることができる。特に、上述のディスク 1 1 0 のように、記録媒体がディスク形状のときには、その構造上、所定のアドレスにアクセスするのに要する時間が比較的大きく、さらに記録容量が大容量になるのに伴い、その時間的な効率は無視できないものとなる。

## 【 0 0 7 4 】

この実施の第 1 の形態によれば、シーケンスページは、1 つのディレクトリを構成するファイル毎の MAC 値と、当該ディレクトリの全ての MAC 値を入力として生成された ICV とからなり、記録媒体上の全てのディレクトリのそれぞれに対応付けられたシーケンスページは、シーケンスブロックにおいてそれぞれフィールドを与えられて格納される。そのため、各シーケンスページで閉じた改竄チェックを行うことが可能である。したがって、上述したような、ディレクトリ毎に同一種類のファイルをまとめて格納する場合にこの発明による改竄チェック方法を適用させることで、時間的に効率的に改竄チェックを行うことができる。

## 【 0 0 7 5 】

次に、この実施の第 1 の形態によるデータ改竄チェック方法の第 2 の応用例について説明する。第 2 の応用例は、著作権保護されたデータと、例えば個人的に撮影や録音したデータや、著作権フリーで配信されたような、著作権保護されていないデータとでディレクトリを分けて記録するような場合に用いて好適な例である。

## 【 0 0 7 6 】

本来、上述のような著作権保護されていないデータは、改竄チェックなどのようなセキュリティのシステムを介さないで複写や移動、更新などを行うことができる。しかしながら、このような著作権保護されていないデータを、上述の改竄チェックの対象としてしまうと、そのデータを再生できなくなったり、場合によっては削除されてしまったりして、ユーザビリティが損なわれてしまう。

## 【 0 0 7 7 】

一方、そのファイルを改竄チェックの対象とするかどうかを、ファイル単位で明示するような方法も考えられる。しかしながら、この方法でも、記録媒体全体で改竄チェック手続を閉じてしまうと、結局、その記録媒体上の全てのファイルをスキミングして、それぞれのファイルについて改竄チェックを行うかどうかを判断しなければならず、上述の第 1 の応用例で述べたのと同様に、時間的なロスが生じてしまう。

## 【 0 0 7 8 】

この発明では、上述したように、データ改竄チェックの対象とするかどうかをディレクトリ単位で判断することができる。したがって、著作権保護が必用なデータと、著作権保護が必用でないデータとを互いに異なるディレクトリに格納することで、時間的なロスを少なく、データ改竄チェックを行うことができる。

## 【 0 0 7 9 】

次に、この発明の実施の第 2 の形態について説明する。この実施の第 2 の形態では、ディスク状記録媒体そのものが有する物理的に固有で且つユニークな情報を用いて MAC 値を生成する。これにより、ファイルを当該ディスク状記録媒体に対して縛ることができ、ファイルの、他の記録媒体への違法な複写などを防止することができる。

## 【 0 0 8 0 】

図 9 は、この実施の第 2 の形態によるデータ改竄チェックを行うための基本的な処理を示す一例の機能ブロック図である。ディスク 2 3 0 は、上述の実施の第 1 の形態と同様に、ユーザデータが記録されるユーザデータ領域 2 3 0 A と、ファイルシステムによる論理アドレスを持たないリードイン領域 2 3 0 B とを有する。さらに、この実施の第 2 の形態では、ディスク 2 3 0 に固有で改竄不可能、若しくは改竄が非常に困難な方法で、メディア固有 ID がディスク 2 3 0 の所定領域に記録されている。

## 【 0 0 8 1 】

メディア固有 ID は、例えばリードイン領域 2 3 0 B などのように、ユーザが簡単にはアクセスできないような領域に記録されるのが好ましい。さらに、メデ

メディア固有IDは、ユーザによる改竄が不可能、若しくは改竄が困難のように、例えば、ディスク230の記録膜そのものを大出力レーザーで破壊する、あるいは、ディスク230の記録面に物理的に傷を付けるといった、破壊系の記録方法で記録されるのがより好ましい。これに限らず、例えばディスク230の出荷時に、スタンプなどによりディスク230の表面や記録面に固有のIDを刻印し、これをメディア固有IDとして用いることもできる。

#### 【0082】

ディスク230のユーザデータ領域230Aに記録されたファイルのそれぞれについて、MACの演算手法に基づき、MAC値が計算される。このとき、上述したメディア固有IDも読み取られ、MAC演算の際の入力として用いられる。図9でいうと、ディスク230に記録されたファイル#1から、著作権情報やファイルの重要情報といったファイルの属性情報と、当該ファイルに固有な鍵となる情報、例えばそのファイルの実データ部分を暗号化する際に用いられたコンテンツ鍵とが読み取られ、MAC演算部231に供給される。一方、ディスク230の例えばリードイン領域230Bに記録されたメディア固有IDが読み取られ、MAC演算部231に供給される。

#### 【0083】

MAC演算部231では、これら、各ファイルから得られたファイルの重要情報およびコンテンツ鍵と、ディスク230に記録されたメディア固有IDとを用いて、MAC値#1を生成する。生成されたMAC値#1は、ファイル#1のヘッダ情報としてファイル#1に格納されると共に、シーケンスブロック232に格納される。シーケンスブロック232は、リードイン領域230Bに記録される。シーケンスブロック232は、上述のシーケンスブロック14に対応し、改竄チェックに関連する情報が格納される領域およびそのデータ構造を指す。

#### 【0084】

図10は、この実施の第2の形態によるデータ改竄チェック方法を示す一例の機能ブロック図である。先ず、図10Aに示されるように、上述の図9で示したのと同様な方法で、ファイル#1上の情報とメディア固有IDとからMAC値#1が生成され、生成されたMAC値#1がファイル#1のヘッダ情報としてフ

イル#1に格納される。また、MAC値#1は、ディスク230上の他のファイルのMAC値と共に、シーケンスブロック232に格納される。ここで、このファイル#1がディスク230とは異なるディスク230'に正規手続きを踏まずに複写あるいは移動された場合を考える。

## 【0085】

ディスク230'には、上述と同様にして、例えばリードイン領域230Bにメディア固有IDが記録されている。なお、ディスク230および230'に記録されているメディア固有IDを、それぞれメディア固有ID-1、メディア固有ID-2と称する。メディア固有IDは、記録メディアに対して固有であって、ディスク230のメディア固有ID-1と、ディスク230'のメディア固有ID-2とは、互いに異なる値となる。

## 【0086】

ファイルが複写あるいは移動されると、図10Bに示されるように、移動先のディスク230'においてデータ改竄チェックが行われる。まず、ディスク230'上のファイル#1のヘッダ情報から、ファイルの重要情報とコンテンツ鍵とが取り出され、MAC演算部231に供給される。それと共に、ディスク230'のメディア固有ID-2が読み出され、MAC演算部231に供給される。MAC演算部231では、供給されたこれらのファイルの重要情報、コンテンツ鍵およびメディア固有ID-2を用いてMAC演算を行い、MAC値#1"を生成する。生成されたMAC値#1"は、比較部233に供給される。

## 【0087】

一方、ファイル#1のヘッダ情報には、複写元のディスク230におけるメディア固有ID-1に基づき生成されたMAC値#1が格納されている。このMAC値#1がファイル#1のヘッダから読み取られ、比較部233に供給される。

## 【0088】

比較部では、供給されたこれらMAC値#1とMAC値#1"とを比較する。上述したように、ディスク230のメディア固有ID-1とディスク230'のメディア固有ID-2とは異なる。そのため、複写元のディスク230上のファイル#1と複写先のディスク230'のファイル#1とが全く同一の内容であっ

ても、メディア固有 I D - 1 を用いて生成された M A C 値 # 1 とメディア固有 I D - 2 を用いて生成された M A C 値 # 1 ” とは異なる値となり、M A C 値の不一致が生じる。

## 【 0 0 8 9 】

さらに、ファイル # 1 のディスク 2 3 0' への複写は、正規の手続きを踏んでいないため、ディスク 2 3 0' 上のシーケンスブロック 2 3 2' には、複写されたファイル # 1 に対応する M A C 値が格納されていない。したがって、上述した図 2 のように、シーケンスブロック 2 3 2' に格納された M A C 値とディスク 2 3 0' 上の全てのファイルの M A C 値とを比較した場合、不正が検出されることになる。

## 【 0 0 9 0 】

このように、この実施の第 2 の形態によれば、異なるディスク間でのファイルの複写や移動は可能であるが、複写または移動先において、不正に複写または移動されたファイルを検出することができる。そのため、例えば複写あるいは移動先のディスクからの当該ファイルの再生を禁止したり、複写あるいは移動先のディスク上からの当該ファイルの削除などを行うことができる。これにより、実質上、あるファイルをあるディスク上に縛ることが可能とされる。

## 【 0 0 9 1 】

また、この実施の第 2 の形態によれば、データ改竄チェックのために、I C V が必ずしも必要ではないことがわかる。I C V は、同一記録媒体上に存在するファイルの整合性を保つためのものであるが、記録媒体上に存在する全ファイルを対象に M A C 演算を行うため、時間的効率が悪かった。しかしながら、この実施の第 2 の形態を用いることで、実質的に、記録媒体とファイルとの対応が一对一になるため、I C V が無くても記録媒体全体での整合性を保つことができる。

## 【 0 0 9 2 】

特に記録媒体がディスク形状のときには、その構造上、所定のアドレスにアクセスするのに要する時間が比較的大きく、さらに記録容量が大容量になるのに伴い、その時間的効率は無視できないものとなる。この実施の第 2 の形態を用いることで、この問題を軽減することができる。

## 【 0 0 9 3 】

次に、この実施の第 2 の形態の変形例について説明する。上述した実施の第 2 の形態では、MAC 演算の際の入力として、記録媒体に固有で、且つ、改竄不可能あるいは改竄困難な ID であるメディア固有 ID を用いていた。これに対して、この変形例では、記録媒体上に記録された、当該記録媒体の欠陥情報であるディフェクト情報を MAC 演算の際の入力として用いている。ディフェクト情報は、十分大きな確率で記録媒体毎に異なった値をとるため、記録媒体固有の情報となり得る。したがって、このディフェクト情報を用いて上述した実施の第 2 の形態と同様の改竄チェックを行うことができる。

## 【 0 0 9 4 】

ディスク状記録媒体には、製造時に必ず、記録領域の物理的な欠陥が発生する。この製造時に発生した記録領域の物理的な欠陥を、ディフェクトと呼ぶ。このディフェクトは、ディスク状記録媒体の製造過程において乱数的に発生し、製造ロットにおいて、あるディスク状記録媒体と全く同一の欠陥状態を持つ他のディスク状記録媒体が存在する確率は、十分に小さい。したがって、このディフェクト情報は、ディスク状記録媒体における固有な物理的情報といえる。

## 【 0 0 9 5 】

この変形例の前提として、ディスク状記録媒体の出荷時には、この欠陥状態がベリファイされ、記録の最小単位毎の欠陥状態が反映された PDL (Primary Defect List) と称されるディフェクト情報が各ディスクに記録される。図 11 は、PDL の一例のデータ構造を示す。PDL は、ビット列からなり、最初の 16 ビットは、このデータが PDL であることを示す固定値とされる。次からの各ビットは、例えばアドレス昇順に準じて、ディスク状記録媒体の最小記録単位のそれぞれを表し、立っている、すなわちビットの値が「1」であるビットに対応した最小記録単位に欠陥が存在し、その最小記録単位が使用不能であることを示す。このビット列が、ディスク状記録媒体の全最小記録単位数にわたり記録される。PDL は、例えばディスク状記録媒体の全記録容量が 2 G B y t e であれば、2 ～ 1 6 K B y t e の容量となる。

## 【 0 0 9 6 】



なお、PDLは、それ自身が改竄されないように、ディスク状記録媒体上の、例えばリードイン領域といった、ユーザにより簡単にアクセスできないような領域に記録されることが必要である。

## 【0097】

この実施の第2の形態の変形例では、このPDLをMAC演算に用いる。これにより、ディスク状記録媒体に記録されたファイルは、当該ディスク状記録媒体において閉じたものとなる。

## 【0098】

図12は、この実施の第2の形態の変形例によるデータ改竄チェックを行うための基本的な処理を示す一例の機能ブロック図である。ディスク240は、上述の実施の第1および第2の形態と同様に、ユーザデータが記録されるユーザデータ領域240Aと、ファイルシステムによる論理アドレスを持たないリードイン領域240Bとを有する。PDLは、リードイン領域240Bに記録される。

## 【0099】

ディスク240のユーザデータ領域240Aに記録されたファイルのそれぞれについて、MACの演算手法に基づき、MAC値が計算される。このとき、上述したPDLも読み取られ、MAC演算の際の入力として用いられる。図12でいうと、ディスク240に記録されたファイル#1から、著作権情報といった重要情報と、当該ファイルに固有な鍵となる情報、例えばそのファイルの実データ部分を暗号化する際に用いられたコンテンツ鍵とが読み取られ、MAC演算部241に供給される。一方、ディスク240の例えばリードイン領域240Bに記録されたPDLが読み取られ、MAC演算部241に供給される。

## 【0100】

MAC演算部241では、これら、各ファイルから得られたファイルの重要情報およびコンテンツ鍵と、ディスク240に記録されたPDLとを用いて、MAC値#1を生成する。生成されたMAC値#1は、ファイル#1のヘッダ情報としてファイル#1に格納されると共に、シーケンスブロック242に格納される。シーケンスブロック242は、リードイン領域240Bに記録される。

## 【0101】

図 1 3 は、この実施の第 2 の形態の変形例によるデータ改竄チェック方法を示す一例の機能ブロック図である。先ず、図 1 3 A に示されるように、上述の図 1 2 で示したのと同様な方法でファイル # 1 上の情報と P D L とから M A C 値 # 1 が生成され、生成された M A C 値 # 1 がファイル # 1 のヘッダ情報としてファイル # 1 に格納される。また、M A C 値 # 1 は、ディスク 2 4 0 上の他のファイルの M A C 値と共に、シーケンスブロック 2 4 2 に格納される。ここで、このファイル # 1 がディスク 2 4 0 とは異なるディスク 2 4 0' に正規手続きを踏まずに複写あるいは移動された場合を考える。

## 【 0 1 0 2 】

ディスク 2 4 0' には、上述と同様にして、例えばリードイン領域 2 4 0 B に P D L が記録されている。なお、ディスク 2 4 0 および 2 4 0' に記録されている P D L を、それぞれ P D L - 1、P D L - 2 と称する。これら P D L - 1 と P D L - 2 とが同一の値となる確率は、上述したように極めて小さい。

## 【 0 1 0 3 】

ファイルが複写あるいは移動されると、図 1 3 B に示されるように、移動先のディスク 2 4 0' においてデータ改竄チェックが行われる。先ず、ディスク 2 4 0' 上のファイル # 1 のヘッダ情報から、ファイルの重要情報とコンテンツ鍵とが取り出され、M A C 演算部 2 4 1 に供給される。それと共に、ディスク 2 4 0' の P D L - 2 が読み出され、M A C 演算部 2 4 1 に供給される。M A C 演算部 2 4 1 では、これらファイルの重要情報、コンテンツ鍵および P D L - 2 を用いて M A C 演算を行い、M A C 値 # 1'' を生成する。生成された M A C 値 # 1'' は、比較部 2 4 3 に供給される。

## 【 0 1 0 4 】

一方、ファイル # 1 のヘッダ情報には、複写元のディスク 2 4 0 における P D L - 1 に基づき生成された M A C 値 # 1 が格納されている。この M A C 値 # 1 がファイル # 1 のヘッダから読み取られ、比較部 2 4 3 に供給される。

## 【 0 1 0 5 】

比較部では、供給されたこれら M A C 値 # 1 と M A C 値 # 1'' とを比較する。上述したように、ディスク 2 4 0 の P D L - 1 とディスク 2 4 0' の P D L - 2

とが一致する確率は、極めて小さいため、複写元のディスク 2 4 0 上のファイル # 1 と複写先のディスク 2 4 0' のファイル # 1 とが全く同一の内容であっても、PDL-1 を用いて生成された MAC 値 # 1 と、PDL-2 を用いて生成された MAC 値 # 1 ” とは、極めて高い確率で異なる値となり、MAC 値の不一致が生じる。

## 【 0 1 0 6 】

さらに、ファイル # 1 のディスク 2 4 0' への複写は、正規の手続きを踏んでいないため、ディスク 2 4 0' 上のシーケンスブロック 2 4 2' には、複写されたファイル # 1 に対応する MAC 値が格納されていない。したがって、シーケンスブロック 2 4 2' に格納された MAC 値とディスク 2 4 0' 上の全てのファイルの MAC 値とを比較した場合、不正が検出されることになる。

## 【 0 1 0 7 】

このように、この実施の第 2 の形態の変形例でも、上述の実施の第 2 の形態と同様に、異なるディスク間でのファイルの複写や移動は可能であるが、複写または移動先において、不正に複写または移動されたファイルを検出することができる。そのため、例えば複写あるいは移動先のディスクからの当該ファイルの再生を禁止したり、複写あるいは移動先のディスク上からの当該ファイルの削除などを行うことができる。これにより、実質上、あるファイルをあるディスク上に縛ることが可能とされる。

## 【 0 1 0 8 】

また、この実施の第 2 の形態の変形例は、上述の実施の第 2 の形態と同様に、データ改竄チェックのために、ICV が必ずしも必要ではないことになる。ICV は、同一記録媒体上に存在するファイルの整合性を保つためのものであるが、記録媒体上に存在する全ファイルを対象に MAC 演算を行うため、時間的効率が悪かった。しかしながら、この変形例を用いることで、実質的に、記録媒体とファイルとの対応が一对一になるため、ICV が無くても記録媒体全体での整合性を保つことができる。

## 【 0 1 0 9 】

特に記録媒体がディスク形状のときには、その構造上、所定のアドレスにアク

セスするのに要する時間が比較的大きく、さらに記録容量が大容量になるのに伴い、その時間的効率は無視できないものとなる。この実施の第 2 の形態の変形例を用いることで、この問題を軽減することができる。

#### 【 0 1 1 0 】

なお、上述の実施の第 1 および第 2 の形態、ならびに、実施の第 2 の形態の変形例は、互いに組み合わせて実施可能なものである。

#### 【 0 1 1 1 】

図 1 4 は、この発明に適用できるディスク状記録媒体 1 の論理フォーマットを、ディスクの形状に対応付けて示す。このディスク状記録媒体 1 の論理フォーマットは、従来例で上述した U D F (Universal Disk Format) に準ずるものである。ディスク状記録媒体 1 (以下、ディスク 1 と称する) において、最内周にリードイン領域 1 0 が配される。リードイン領域 1 0 の外側から論理セクタ番号 (L S N : Logical Sector Number) が割り当てられ、順に、ボリューム情報領域 1 1、領域 D A N (Data Area Number) - 1、D A N - 2、D A N - 3 およびボリューム情報領域 1 2 が配され、最外周にリードアウト領域 1 3 が配される。領域 D A N - 1 ~ D A N - 3 には、論理ブロック番号が割り当てられる。

#### 【 0 1 1 2 】

図 1 5 は、ボリューム情報領域 1 1 および 1 2 の一例の内容を示す。ボリューム情報領域 1 1 には、U D F の規定に基づき、V R S (Volume Recognition Sequence)、M V D S (Main Volume Descriptor) および L V I S (Logical Volume Integrity Sequence) が書き込まれる。ボリューム情報領域 1 1 の終端には、アンカーポイントが置かれる。また、ボリューム情報領域 1 1 の内容は、リードアウト領域 1 3 の内側のボリューム情報領域 1 2 に R V D S (Reserve Volume Descriptor Sequence) として 2 度書きされる。ボリューム情報領域 1 2 の先頭および終端には、アンカーポイントが置かれる。ボリューム情報領域 1 2 の終端のアンカーポイントは、最終論理セクタ番号に対応する。

#### 【 0 1 1 3 】

論理セクタ番号が 2 7 2 から (最終論理セクタ番号 - 2 7 2) の間は、L V S (Logical Volume Space) とされ、パーティション領域が設けられる。この L V S

に、上述の領域DAN-1～DAN-3が配される。LVSの最内周側に設けられる領域DAN-1は、UDFの規定に基づくFSD(File Set Descriptor)およびSBD(Space Bitmap Descriptor)からなる。SBDは、ディスク1の全体の空きエリア情報を、セクタ毎にフラグを立てることで表現する。また、領域DAN-1には、ファイルシステムの階層構造のルートディレクトリのFEのアドレスが示される。

## 【0114】

領域DAN-2は、ディレクトリのFE(File Entry)と、その実体のFID(File ID)とが置かれる領域である。すなわち、これらディレクトリのFEとその実体のFIDとは、領域DAN-2にまとめて記録されることになる。領域DAN-2は、後述するフォーマット時に、予め所定の容量が連続的に確保される。詳細は後述するが、領域DAN-2の未使用領域は、特定の属性が付されたファイルとして確保される。以下、この領域DAN-2の未使用領域からなるファイルを、EIF(Entry Information File)と称する。未使用領域をEIFとしてファイル化して扱うことで、上述のSBDにおいて、この未使用領域が空きエリアとして認識されないようにできる。

## 【0115】

なお、従来例で既に述べたが、FEは、ファイルやディレクトリの実体の場所(アドレス)および大きさを示す。FE中のAD(Allocation Descriptor)によって、これらの情報が記される。また、FIDは、ファイルやディレクトリの名前と、FEの場所(アドレス)および大きさを示す。FID中のICB(Information Control Block)によってこれらの情報が記される。

## 【0116】

領域DAN-3には、ファイルのFEとその実体とが置かれる領域である。領域DAN-3において、ファイルのFEとそのFEに対応したファイルは、アドレス的に連続して配置される。ファイルを追加する際には、既存のファイルに対してアドレス的に連続的に、追加されるファイルのFEが配置され、さらに、アドレス的に連続してファイルの実体が配置される。このように、ファイルのFEおよび実体をアドレス的に連続して配置することにより、ファイルへのアクセス

を高速に行うことができる。

【0117】

図16および図17を用いて、このディスク状記録媒体1におけるディレクトリ、ファイルおよび空きエリアの管理方法について説明する。図16は、上述の図14に対して、領域DAN-1～DAN-3を抜き出した図である。ここでは、データの記録方向は、図16に一例が示されるように、反時計回りであるものとする。図17は、各FE、FIDおよび実体の一例の階層構造を示す。

【0118】

例えば、ルートディレクトリのFEがLSN=aから開始されとする。ルートディレクトリのFE中のADによって、ルートディレクトリの実体のアドレスおよび大きさが示される。ルートディレクトリのアドレスは、ルートディレクトリのFEと連続的に配置できるようにされており、例えばLSN=a+1とされる。ルートディレクトリの実体は、1以上のFIDを含む。ルートディレクトリのサブディレクトリ（以下、サブディレクトリと略称する）のFEの名前、アドレスおよび大きさがFIDに記される。サブディレクトリのFEは、ルートディレクトリの実体と連続的になるように配置され、例えばLSN=a+2とされる。このサブディレクトリのFE中のADによって、当該サブディレクトリの実体のアドレスおよび大きさが記される。このサブディレクトリの実体のアドレスは、当該サブディレクトリのFEと連続的になるように配置され、例えばLSN=a+3とされる。サブディレクトリに実体は、1以上のFIDを含み、ファイルや他のサブディレクトリのFEの名前、アドレスおよび大きさが記される。

【0119】

各FE、FIDおよび実体がこのように参照されることで、図17に一例が示されるように、領域DAN-2の最内周の所定位置に配置されたルートディレクトリのFEに対して連続的に、ルートディレクトリの実体およびルートディレクトリのサブディレクトリ情報などが配置される。

【0120】

一方、図17を参照し、ルートディレクトリの実体中のFIDによって、EIFのFEの名前、アドレスおよび大きさが記される。そして、EIFのFE中の

ADによって、EIFの実体のアドレスおよび大きさが記される。このように、EIFはファイルとして扱われるので、他のファイルと同様に、FEによってそのアドレスおよび大きさが示される。

## 【0121】

EIFのFEは、図16に一例が示されるように、例えばEIFの実体よりも後ろに配置される。EIFの実体の開始および／または終了アドレス、ならびに、大きさは、後述するように、領域DAN-2に書き込まれる各情報の量によって変動する。

## 【0122】

以上、ルートディレクトリのFE、ルートディレクトリの実体、ルートディレクトリのサブディレクトリのFE、ルートディレクトリのサブディレクトリの実体、EIFのFEおよびEIFの実体は、領域DAN-2に配置される。

## 【0123】

領域DAN-3には、ファイルのFEおよびファイルの実体が配置される。ファイルの実体は、実際にユーザデータなどが書き込まれる領域である。図17に一例が示されるように、ルートディレクトリの実体中のFIDによって名前、アドレスおよび大きさが記されたファイルのFEは、領域DAN-3に配置される。このときの、ファイルのFEの開始アドレスをLSN=dとする。ファイルのFE中のADによって、当該ファイルの実体のアドレスおよび大きさが示される。ファイルの実体は、当該ファイルのFEと連続的になるように配置され、例えば開始アドレスがLSN=d+1とされる。

## 【0124】

上述したように、領域DAN-2は、このディスク1のフォーマット処理時に予め確保される。次に、このディスク1の一例のフォーマット方法について、概略的に説明する。なお、リードイン領域10およびリードアウト領域13は、例えばディスク1の製造のプレス行程の際に予め作成されるなどして、フォーマット処理以前から既に存在するものとする。フォーマット処理は、ディスク1の内周側から外周側にかけて進められる。

## 【0125】

フォーマット処理が開始されると、最初に、上述したVRS、MVDSおよびLVISがリードイン領域10の外側から書き込まれる。次に、LVISが作成される。LVISにおいて、まず、領域DAN-1が作成される。FSDが書き込まれ、ルートディレクトリの位置が決められる。そして、SBDが作成される。このときに、上述したEIFの領域をSBDにおいて使用済み領域とすることで、EIFの領域が確保される。

## 【0126】

SBDが作成され領域DAN-1が作成されると、次に、領域DAN-1の外側から領域DAN-2が作成される。領域DAN-2の作成において、まず、領域DAN-1で書き込まれたFSDに基づき、所定アドレスにルートディレクトリFEおよびルートディレクトリの実体が連続的に書き込まれる。次に、作成されたルートディレクトリの実体に、EIFのFIDが追加される。このFIDにおいて、EIFのFEのアドレスが指定される。

## 【0127】

このとき、EIFの属性がFID中に指定される。指定されるEIFの属性は、EIFが他の機器やOS (Operating System)によって消去、書き換え、移動などが行われなくするためのものである。例えば「隠しファイル属性」、「システムファイル属性」および「読み出し専用ファイル属性」が、共にEIFの属性として指定される。

## 【0128】

「隠しファイル属性」は、この属性が設定されたファイルを通常の方法では閲覧できなくする属性である。「システムファイル属性」は、この属性が設定されたファイルがシステムのために必要なファイルであることを示す属性である。「読み出し専用ファイル属性」は、この属性が設定されたファイルが読み出し専用であって、変更や消去がシステムによって禁止されることを示す属性である。これら3つの属性を共にファイルに指定することで、意図的な操作以外には、そのファイルに対する消去、書き換え、移動などの処理を行うことができなくされる。なお、これらの属性は、所定の方法で解除することができる。

## 【0129】



次に、E I FのF Eが作成される。上述したように、F Eでは、当該ファイルのアドレスと大きさが指定される。したがって、F Eを指定するだけで、当該ファイルが存在することになり、ダミーファイルとして用いることができる。E I FのF Eには、「読み出し専用ファイル属性」および「システムファイル属性」が指定される。

## 【 0 1 3 0 】

このように、領域D A N - 2内にE I Fを存在させることで、領域D A N - 2の空きエリアをE I Fによって確保することができる。上述したように、D A N - 2には、フォーマット処理後に、サブディレクトリのF Eおよび実体書き込まれる。このときには、E I Fの領域を削って、これらサブディレクトリのF Eおよび実体が領域D A N - 2に作成される。

## 【 0 1 3 1 】

なお、詳細は後述するが、領域D A N - 2の作成順は、上述の順序に限られず、他の順序で行うようにしても良い。このとき当然、領域D A N - 2における各情報の配置順序も変わってくる。

## 【 0 1 3 2 】

このようにして領域D A N - 2が作成される。領域D A N - 2の外側は領域D A N - 3であるが、領域D A N - 3では、特に何も処理が行われない。例えば、領域D A N - 3として指定される領域を飛び越して次の処理がなされる。領域D A N - 3の次は、R V D Sが作成される。これは、上述したように、先に作成されたV R S、M V D SおよびL V I Sの情報が2度書きされる。R V D Sが作成されて、ディスク1のフォーマット処理が完了される。

## 【 0 1 3 3 】

図18は、この発明に適用することができるドライブ装置の一例の構成を示す。ここでは、上述したディスク1を記録層に相変化金属材料を用いたものとし、ドライブ装置は、レーザの出力を調節することで記録層に加える温度を制御して結晶／非結晶に状態を変えさせる相変化技術によって、ディスク1にデータの記録を行うものとする。

## 【 0 1 3 4 】

ディスク 1 は、スピンドルモータ 2 2 によって、回転駆動される。ディスク 1 にデータを記録し、また、データをディスク 1 から再生するために、光ピックアップ 2 3 が設けられている。光ピックアップ 2 3 が送りモータ 2 4 によってディスク径方向に送られる。

#### 【 0 1 3 5 】

外部のホストコンピュータ 3 0 からのデータがインターフェイス 2 9 (例えば S C M S (Serial Copy Management System) ) を介してドライブに供給される。インターフェイス 2 9 には、エンコーダ／デコーダブロック 2 5 が接続され、エンコーダ／デコーダブロック 2 5 には、バッファメモリ 2 6 が接続されている。バッファメモリ 2 6 は、ライトデータまたはリードデータを保持する。

#### 【 0 1 3 6 】

ライトデータがインターフェイス 2 9 を介してエンコーダ／デコーダブロック 2 5 に供給される。エンコーダ／デコーダブロック 2 5 では、記録時には、上述したフォーマットのデータを生成し、次にそのフォーマットに従ってデータをエンコードする。再生時には、デコード処理を行い、デジタルデータをインターフェイス 2 9 を介してホストコンピュータ 3 0 に出力する。アドレスは、例えばエンコーダ／デコーダブロック 2 5 において、サブコードとして付加され、また、データ中のヘッダに対しても付加される。

#### 【 0 1 3 7 】

エンコーダ／デコーダブロック 2 5 からの記録データが記録イコライザ 2 7 を介してレーザドライバ 2 8 に供給される。レーザドライバ 2 8 では、ディスク 1 に対して記録データを記録するための所定のレベルを有するドライブ波形が生成される。レーザドライバ 2 8 の出力が光ピックアップ 2 3 に対して供給され、データが記録される。レーザドライバ 2 8 は、R F 信号処理ブロック 3 1 内の A P C (Automatic Power Control) によって、上述したように、レーザパワーが適切なものに制御される。また、ディスク 1 からの戻り光により発生した信号が R F 信号処理ブロック 3 1 に供給される。アドレス抽出回路 3 2 では、R F 信号処理ブロックから供給された信号に基づき、アドレス情報の抽出を行う。抽出されたアドレス情報は、後述する制御用マイコン 3 3 に供給される。

## 【 0 1 3 8 】

また、RF信号処理ブロック31では、マトリックスアンプがフォトディテクタの検出信号を演算することによって、トラッキングエラー信号TERR、フォーカスエラー信号FERRを生成する。トラッキングエラー信号、フォーカスエラー信号がサーボブロック34に供給される。

## 【 0 1 3 9 】

制御用マイコン33がアドレスを使用してシーク動作を制御し、また、制御信号を使用してレーザパワーの制御等を行う。制御用マイコン33は、CPU(Central Processing Unit)、RAM(Random Access Memory)およびROM(Read Only Memory)などからなり、インターフェイス29、エンコーダ/デコーダブロック25、RF信号処理ブロック31、サーボブロック34等、ドライブの全体を制御する。また、制御用マイコン33に対してメモリ36が続される。

## 【 0 1 4 0 】

さらに、制御用マイコン33によって、ディスク1のリードイン領域10に対するアクセスが制御される。上述した、リードイン領域内に記録されるシーケンスブロックが制御用マイコン33によって読み出される。読み出されたシーケンスブロックは、例えばメモリ36に記憶される。また、ディスク1のユーザデータ領域(領域DAN-3)に記録されたファイルのヘッダ情報が制御用マイコン33によって読み取られ、例えばメモリ36に記憶される。また、実施の第2の形態およびその変形例に示されるような、ディスク1に対して物理的に記されたメディア固有IDやPDLも、制御用マイコン33によって読み取られ、読み取られた情報がメモリ36に記憶される。

## 【 0 1 4 1 】

上述したMAC演算部や比較部は、例えば制御用マイコン33においてソフトウェア的に構成される。勿論、MAC演算部や比較部をハードウェア的に別途、設けてもよい。MAC演算部により、メモリ36に記憶されたこれらの情報に基づき上述したようにMAC値などが生成され、データ改竄チェックなどの処理がなされる。

## 【 0 1 4 2 】

ディスク 1 を再生することで得られる R F 信号がエンコーダ／デコーダブロック 2 5 に供給され、エンコーダ／デコーダブロック 2 5 では、記録時に施された変調処理の復調、エラー訂正符号の復号（すなわち、エラー訂正）等の所定のフォーマットに準ずるデコードを行う。エンコーダ／デコーダブロック 2 5 では、再生データがバッファメモリ 2 6 に格納される。ホストコンピュータ 3 0 からのリードコマンドが受け付けられると、リードデータがインターフェイス 2 9 を介してホストコンピュータ 3 0 に対して転送される。

## 【 0 1 4 3 】

R F 信号処理ブロック 3 1 からのフレーム同期信号、トラッキングエラー信号およびフォーカスエラー信号と、アドレス抽出回路からのアドレス情報がサーボブロック 3 4 に供給される。サーボブロック 3 4 は、光ピックアップ 2 3 に対するトラッキングサーボおよびフォーカスサーボと、スピンドルモータ 2 2 に対するスピンドルサーボと、送りモータ 2 4 に対するスレッドサーボを行う。

## 【 0 1 4 4 】

なお、上述では、ドライブ装置に対してホストコンピュータ 3 0 が接続されると説明したが、これはこの例に限定されない。ドライブ装置に接続される機器は、デジタル信号の入出力を行いインターフェイスが適合していれば、他の機器でも良い。例えば、このドライブ装置は、例えば撮像画像をディスク状記録媒体に記録するようにされたカメラ付き携帯用デジタルビデオレコーダに内蔵されるものとしても良い。

## 【 0 1 4 5 】

上述では、ディスク 1 に対するフォーマットデータをエンコーダ／デコーダブロック 2 5 で生成するように説明したが、これはこの例に限定されない。フォーマットデータは、制御用マイコン 3 3 で生成することができる。また、フォーマットデータは、ホストコンピュータ 3 0 から供給するようにしても良い。

## 【 0 1 4 6 】

## 【発明の効果】

以上説明したように、この発明によれば、ファイル毎の M A C 値をディレクトリ単位でまとめて管理しているために、アプリケーションの仕様に応じて、デー

タ改竄チェックに要する手続を最小限化できるという効果がある。特に、記録媒体にディスク状記録媒体を用いた場合に、ファイル毎のデータ改竄チェックを行う際のアクセス性を大幅に向上できる効果がある。

## 【0147】

また、この発明によれば、データ改竄チェックを行うデータ、すなわち、著作権を保護するデータがディレクトリ単位でまとめられるため、著作権の保護が必要ないデータに対するアクセス性を確保できる効果がある。またそのため、データ改竄チェック管理スペース、すなわちシーケンスブロックのサイズも最小限のものとすることができる効果がある。

## 【0148】

さらに、改竄チェック値（MAC値）のリストからある特手kのファイルに対応する値を検索する際に、対象となるディレクトリに対応したシーケンスページ内のMACエン트리数が検索の母数となるため、ディスク全体のMACエントリを母数とする場合に比べて検索効率を改善できる効果がある。

## 【0149】

さらにまた、改善チェック管理スペースであるシーケンスブロックのサイズが可変長とされているため、シーケンスブロックのサイズをファイルやディレクトリ数に応じた最小限のものにすることができ、スペース効率と検索効率とを改善できる効果がある。

## 【図面の簡単な説明】

## 【図1】

ディスク記録媒体に新たにファイルが追加されたときの改竄チェック方法を概略的に示す一例の機能ブロック図である。

## 【図2】

ディスク上である特定のファイルを再生／移動する場合や、システムにおいて改竄チェックが要求されたタイミングにおける一例の手続を示す機能ブロック図である。

## 【図3】

実施の一形態による改竄チェック値のデータ構造を概略的に示す略線図である

【図 4】

ディスクの一例の物理フォーマットを部分的に示す略線図である。

【図 5】

シーケンスブロックの一例の論理フォーマットを示す略線図である。

【図 6】

シーケンスページの一例の論理フォーマットを示す略線図である。

【図 7】

実施の一形態によるシーケンスブロックの更新手続きを示す略線図である。

【図 8】

救済措置を含めたシーケンスブロックの更新手続きの一例の処理を示すフローチャートである。

【図 9】

実施の第 2 の形態によるデータ改竄チェックを行うための基本的な処理を示す一例の機能ブロック図である。

【図 1 0】

実施の第 2 の形態によるデータ改竄チェック方法を示す一例の機能ブロック図である。

【図 1 1】

P D L の一例のデータ構造を示す略線図である。

【図 1 2】

実施の第 2 の形態の変形例によるデータ改竄チェックを行うための基本的な処理を示す一例の機能ブロック図である。

【図 1 3】

実施の第 2 の形態の変形例によるデータ改竄チェック方法を示す一例の機能ブロック図である。

【図 1 4】

この発明に適用できるディスク状記録媒体の論理フォーマットをディスクの形状に対応付けて示す略線図である。

【図 1 5】

ボリューム情報領域の一例の内容を示す略線図である。

【図 1 6】

ディスク状記録媒体でのディレクトリ、ファイルおよび空きエリアの管理方法について説明するための略線図である。

【図 1 7】

ディスク状記録媒体でのディレクトリ、ファイルおよび空きエリアの管理方法について説明するための略線図である。

【図 1 8】

この発明に適用することができるドライブ装置の一例の構成を示すブロック図である。

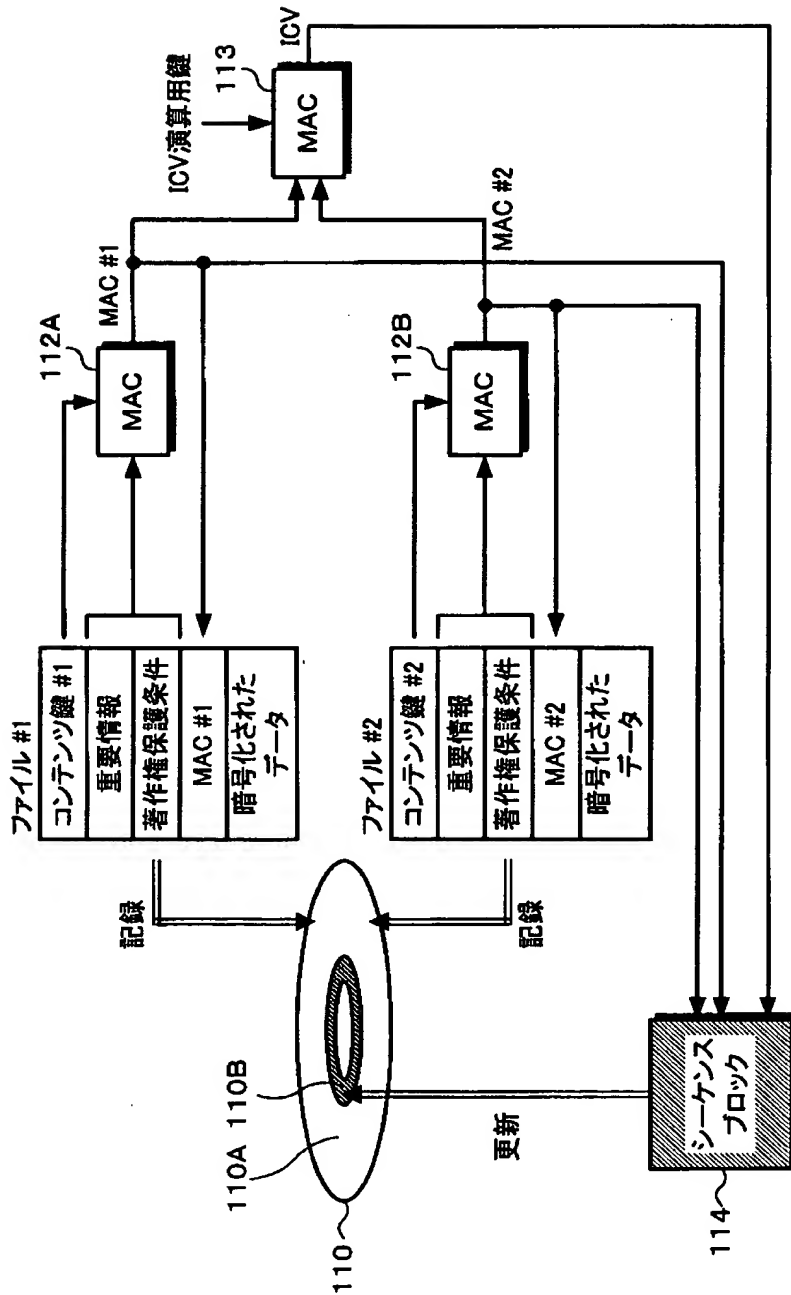
【符号の説明】

1・・・ディスク状記録媒体、10・・・リードイン領域、33・・・制御用マイコン、36・・・メモリ、114, 232, 242・・・シーケンスブロック

【書類名】

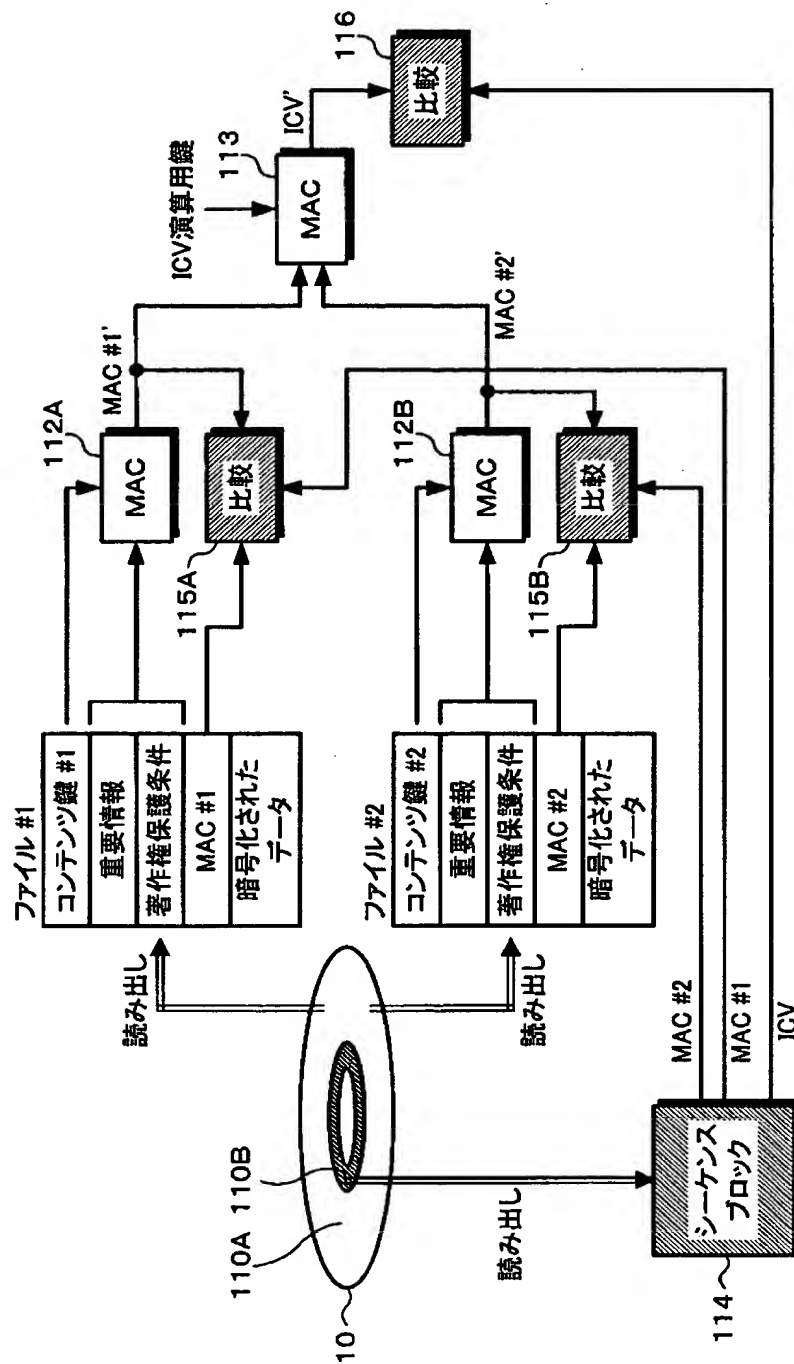
凶面

【図 1】

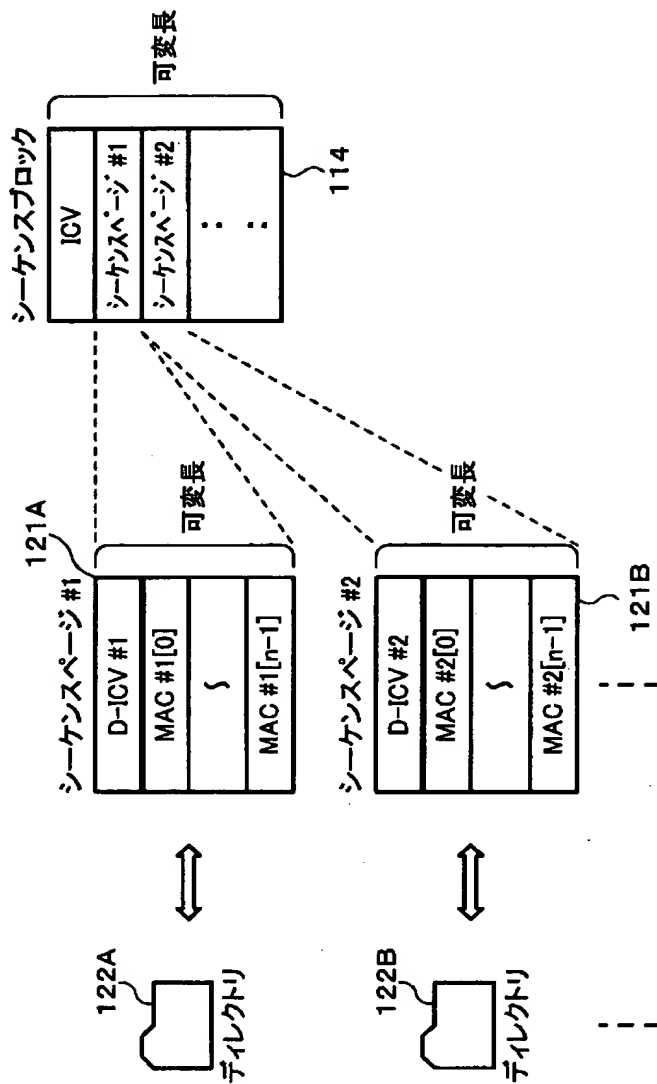




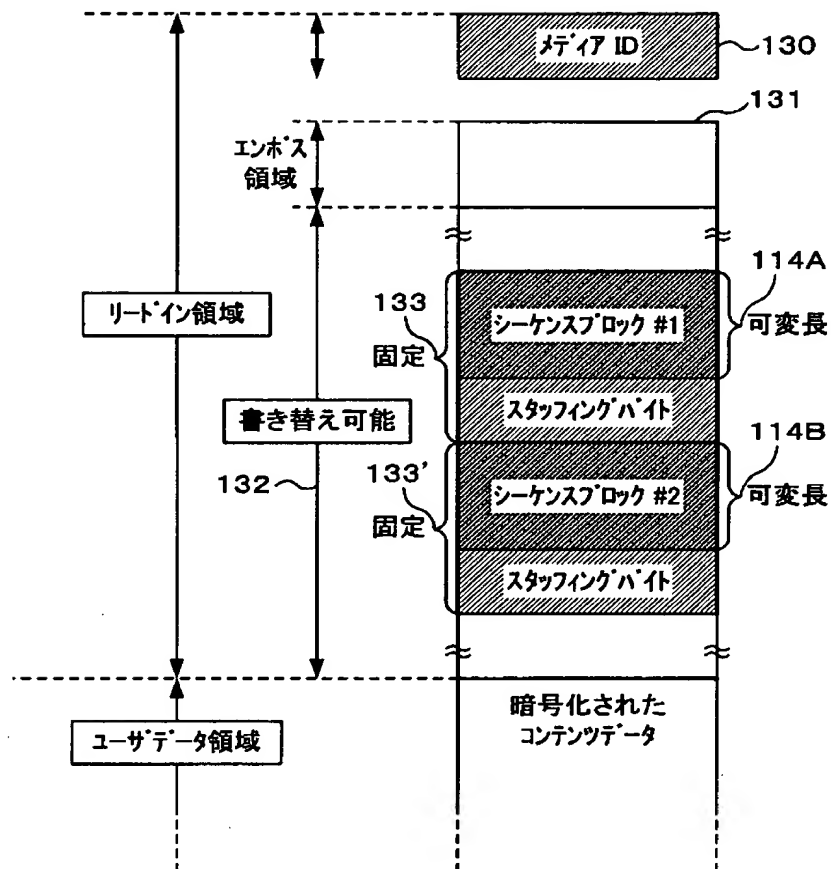
【図 2】



【図 3】



【図 4】



【図 5】

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
SPE Num		Block Size(Byte Count)				Revision				Reserved					
Reserved															
シーケンスページエントリ [0]															
シーケンスページエントリ [1]															
:															
シーケンスページエントリ [m-1]															
スタッキング バイト															
0x000XXXXX															
0x0001FFF0															

SPE Num: Sequence Page Entry Number  
シーケンスページの総エントリー数

Block Size: Sequence Block Size  
シーケンスブロックのサイズ、先頭バイトから最終エントリーの最終バイトまで  
バイト数カウント

Revision: Revision Number  
シーケンスブロックの書き換え回数、有効/無効の状態  
初期状態0から1インクリメント  
0xFFFFFFFF = Invalid Number

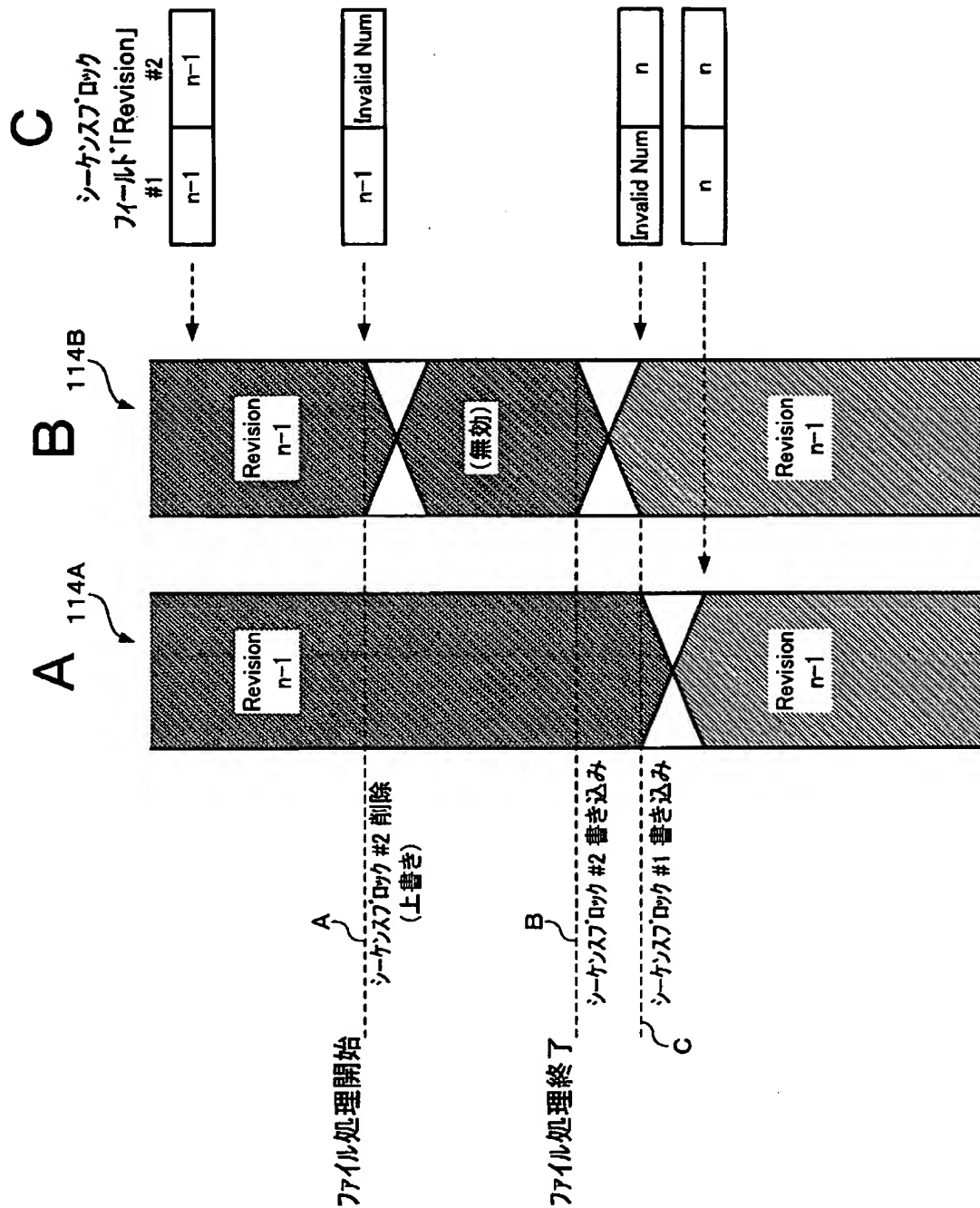
このシーケンスブロックは無効、もしくは書き換え途中であることを示す

【図 6】

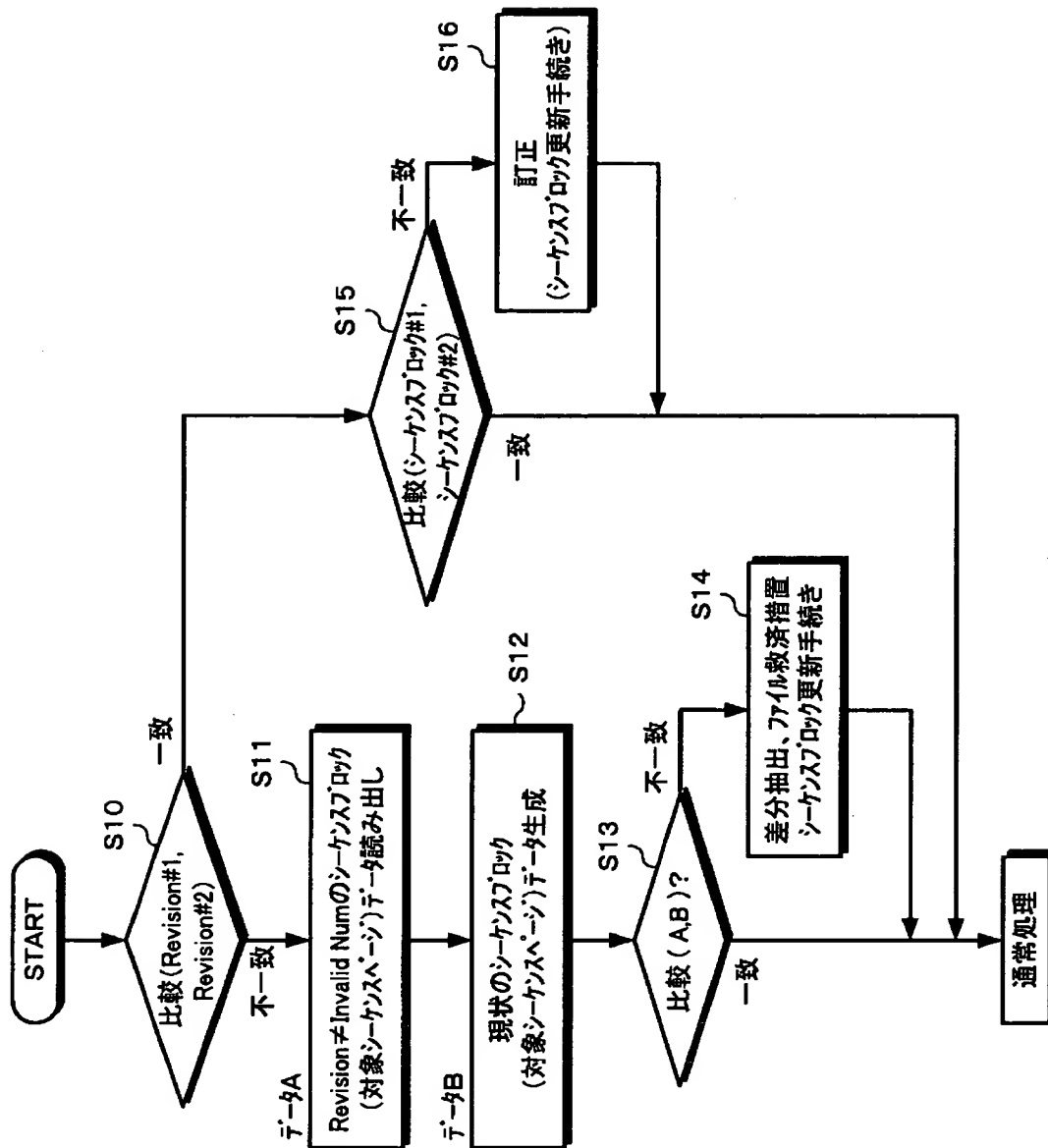
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00000000		Page ID	Entry Num	Block Size(バイトカウント)				Reserved							
0x00000010				D-ICV				C_MAC[0]							
								:							
(0x0001D4C0)				C_MAC[n-2]				C_MAC[n-1]							

- Page ID: Sequence Page ID  
シーケンスページとフォルダを関連付けるためのID
- Entry Num: MAC Entry Number  
総エントリ数
- Page Size: Sequence Page Size  
シーケンスページのサイズ、先頭バイトから最終エントリーの最終バイトまで  
バイト数カウント
- C\_MAC[n]: Contents MAC Value  
各ファイル(コンテンツ)毎に算出されたMAC値

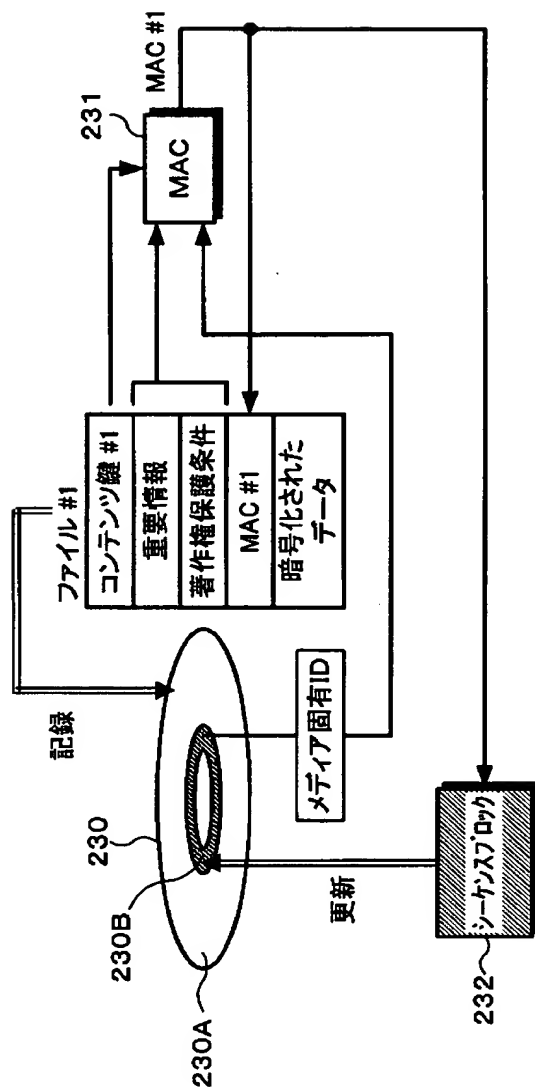
【図 7】



【図 8】

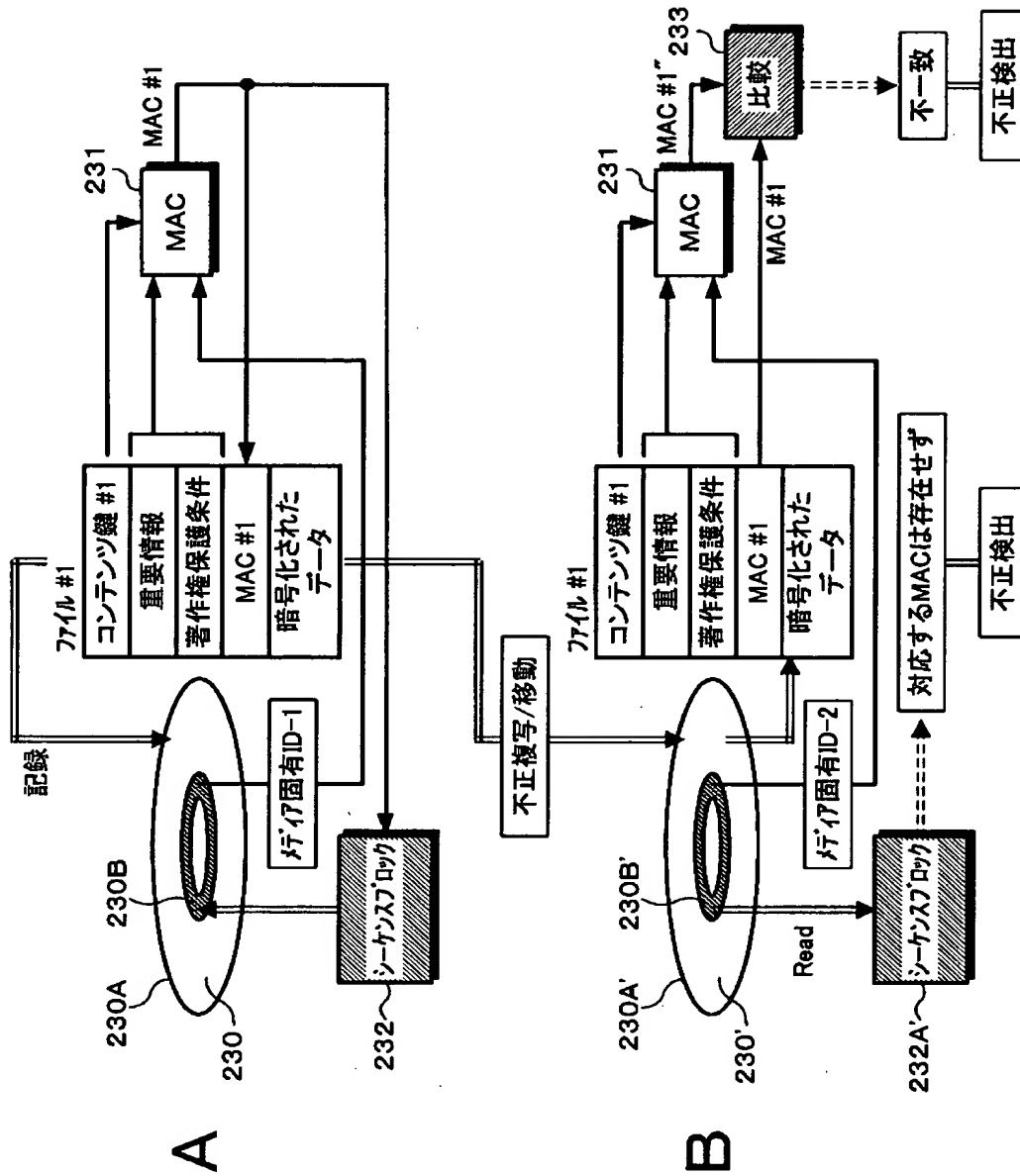


【図9】

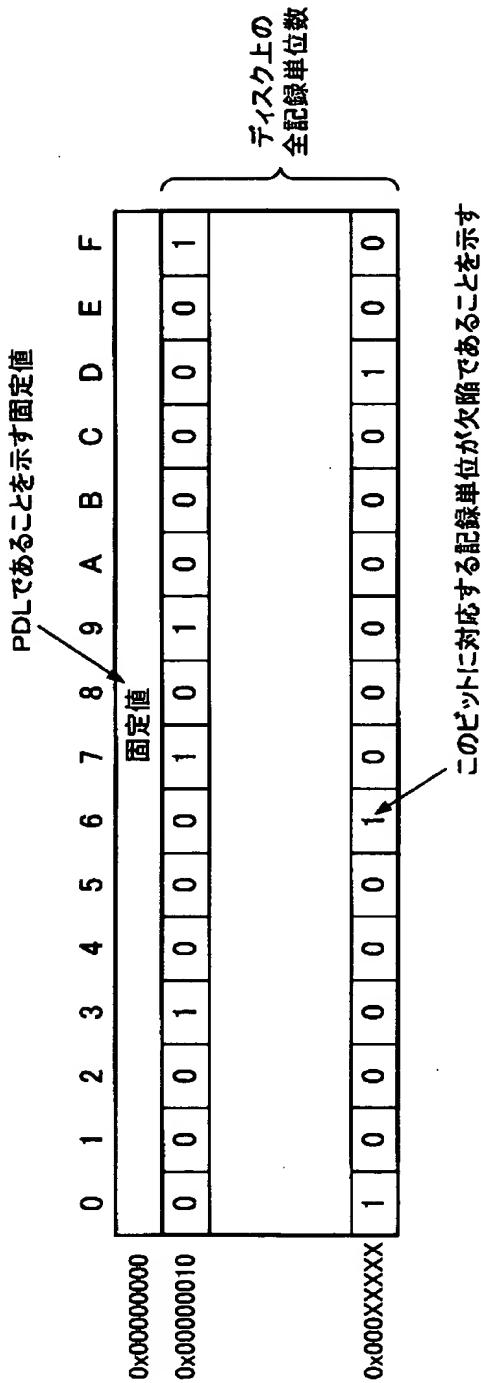




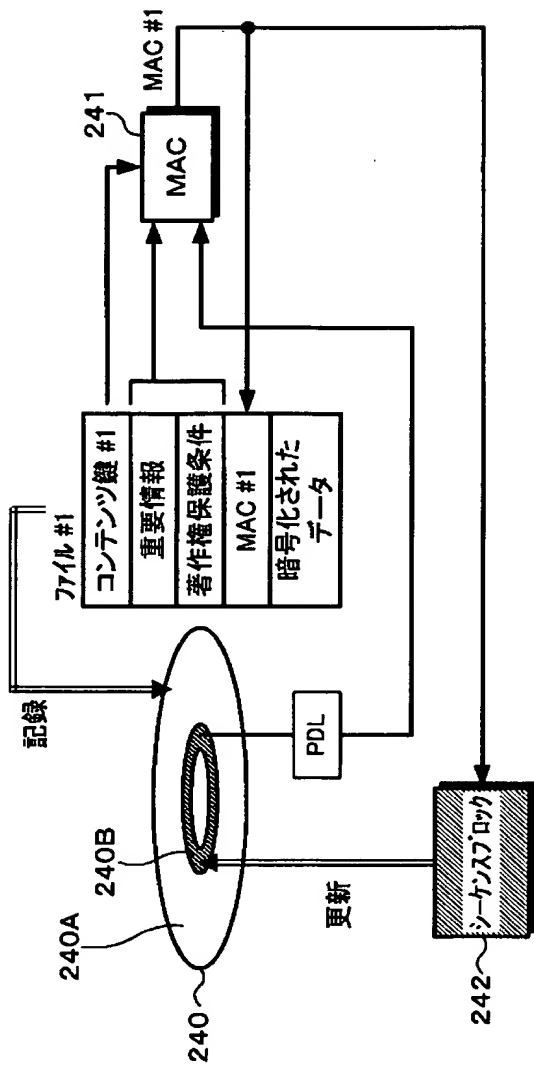
【図10】



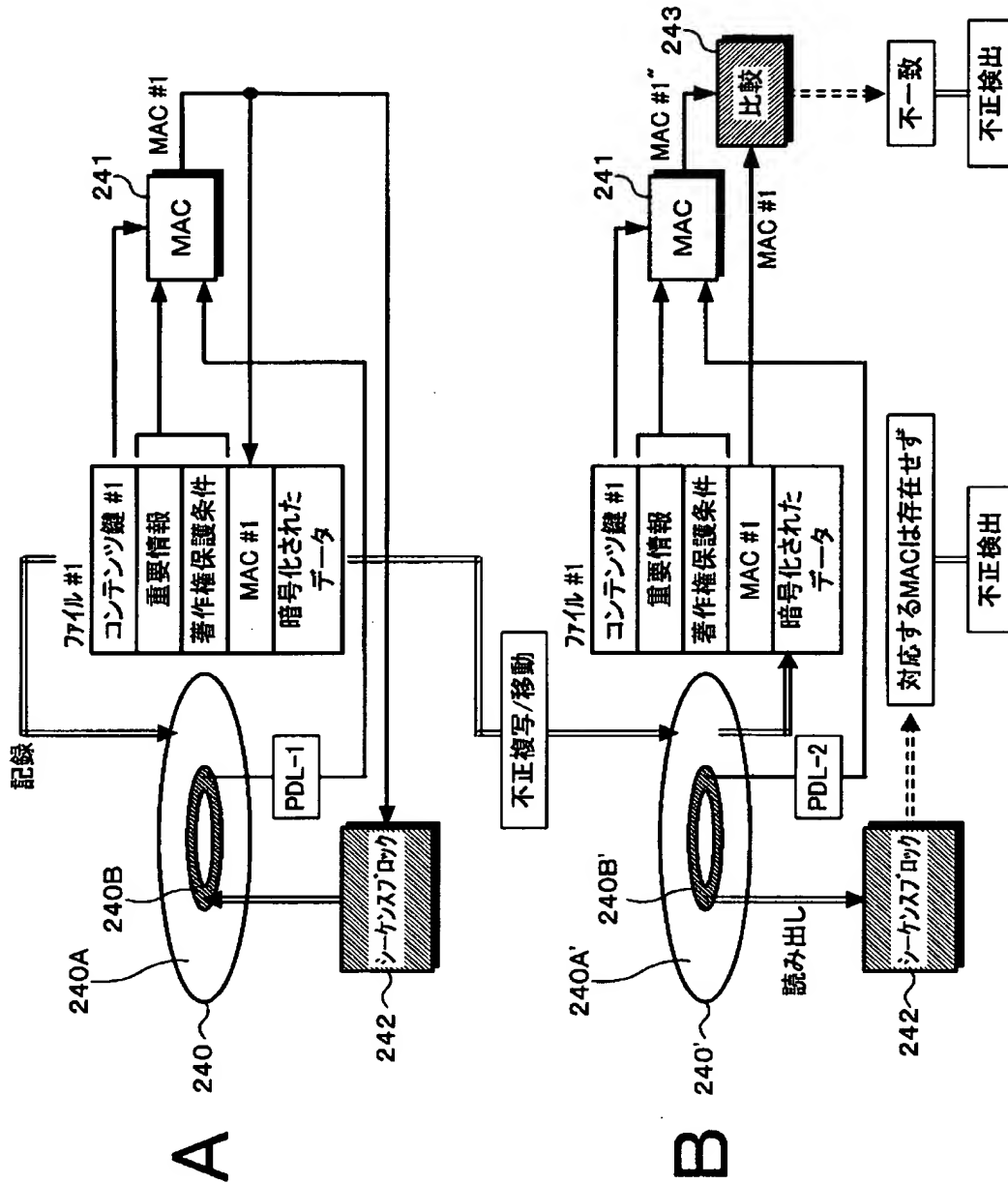
【図 1 1】



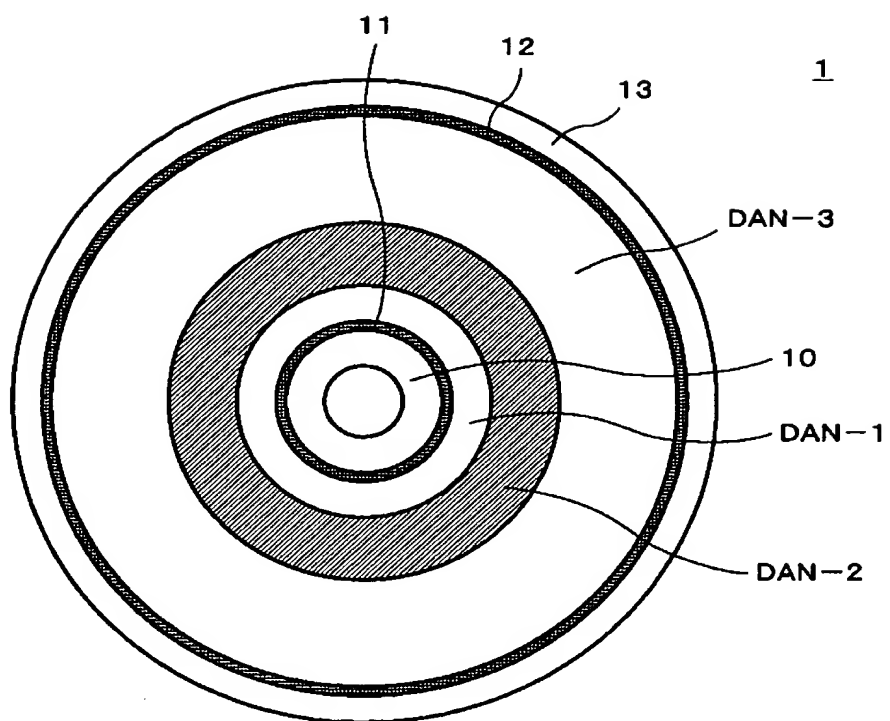
【図 1 2】



【图 1 3】



【図14】



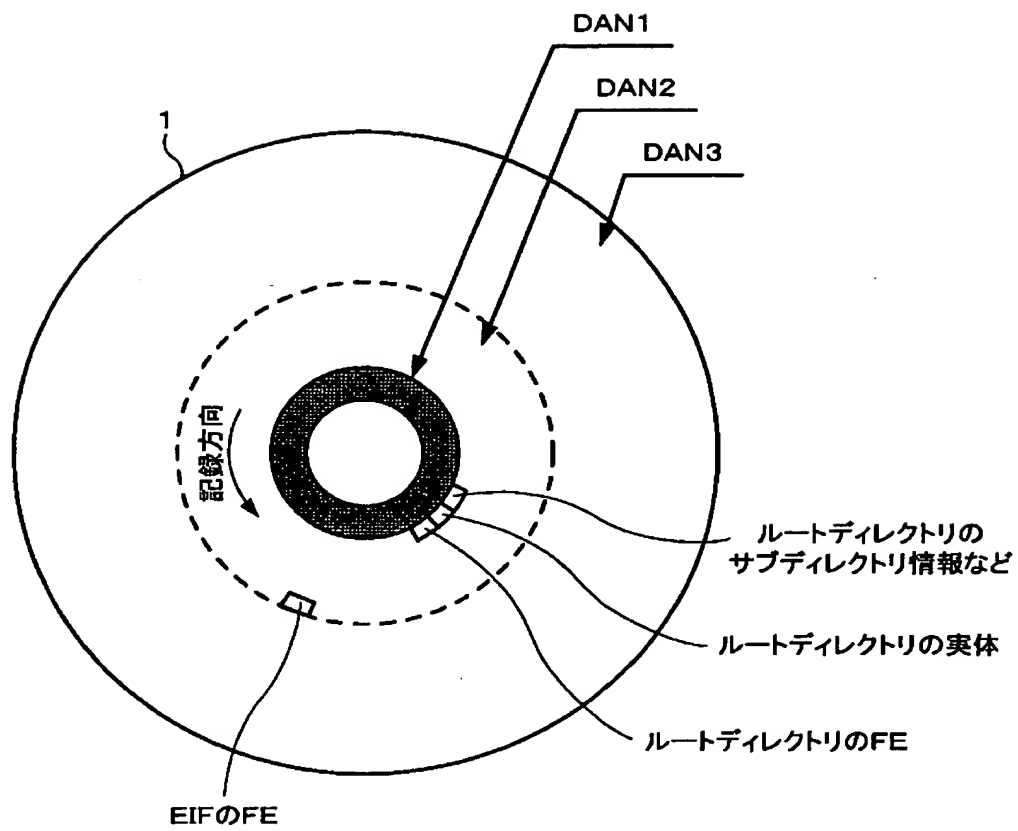
【図 1 5】

LSN	Description	Structure	LBN
0 to 15	Reserved (all 00h bytes)		Not Assigned
16	Beginning Extended Area Descriptor	Volume Recognition Sequence (VRS)	
17	NSR Descriptor		
18	Terminating Extended Area Descriptor		
19 to 31	Reserved (all 00h bytes)		
32	Primary Volume Descriptor	Main Volume Descriptor Sequence (MVDS)	
33	Implementation Use Volume Descriptor		
34	Partition Descriptor		
35	Logical Volume Descriptor		
36	Unallocated Space Descriptor		
37	Terminating Descriptor		
38 to 47	Trailing Logical Sectors (all 00h bytes)		
48	Logical Volume Integrity Descriptor	Logical Volume Integrity Sequence (LVIS)	
49	Terminating Descriptor		
50 to 63	Trailing Logical Sectors (all 00h bytes)		
64 to 255	Reserved (all 00h bytes)		
256	Anchor Volume Descriptor Pointer	First Anchor Point	
257 to 271	all 00h bytes Data		
272 to Last LSN-272	Descriptors for File Structure and Files	Partition (LVS)	0 to Last LBN
Last LSN-271 to Last LSN-257	all 00h bytes Data		Not Assigned
Last LSN-256	Anchor Volume Descriptor Pointer	Second Anchor Point	
Last LSN-255 to Last LSN-224	Reserved (all 00h bytes)		
Last LSN-223	Primary Volume Descriptor	Reserve Volume Descriptor Sequence (RVDS)	
Last LSN-222	Implementation Use Volume Descriptor		
Last LSN-221	Partition Descriptor		
Last LSN-220	Logical Volume Descriptor		
Last LSN-219	Unallocated Space Descriptor		
Last LSN-218	Terminating Descriptor		
Last LSN-217 to Last LSN-208	Trailing Logical Sectors (all 00h bytes)		
Last LSN-207 to Last LSN-1	Reserved (all 00h bytes)		
Last LSN	Anchor Volume Descriptor Pointer	Third Anchor Point	

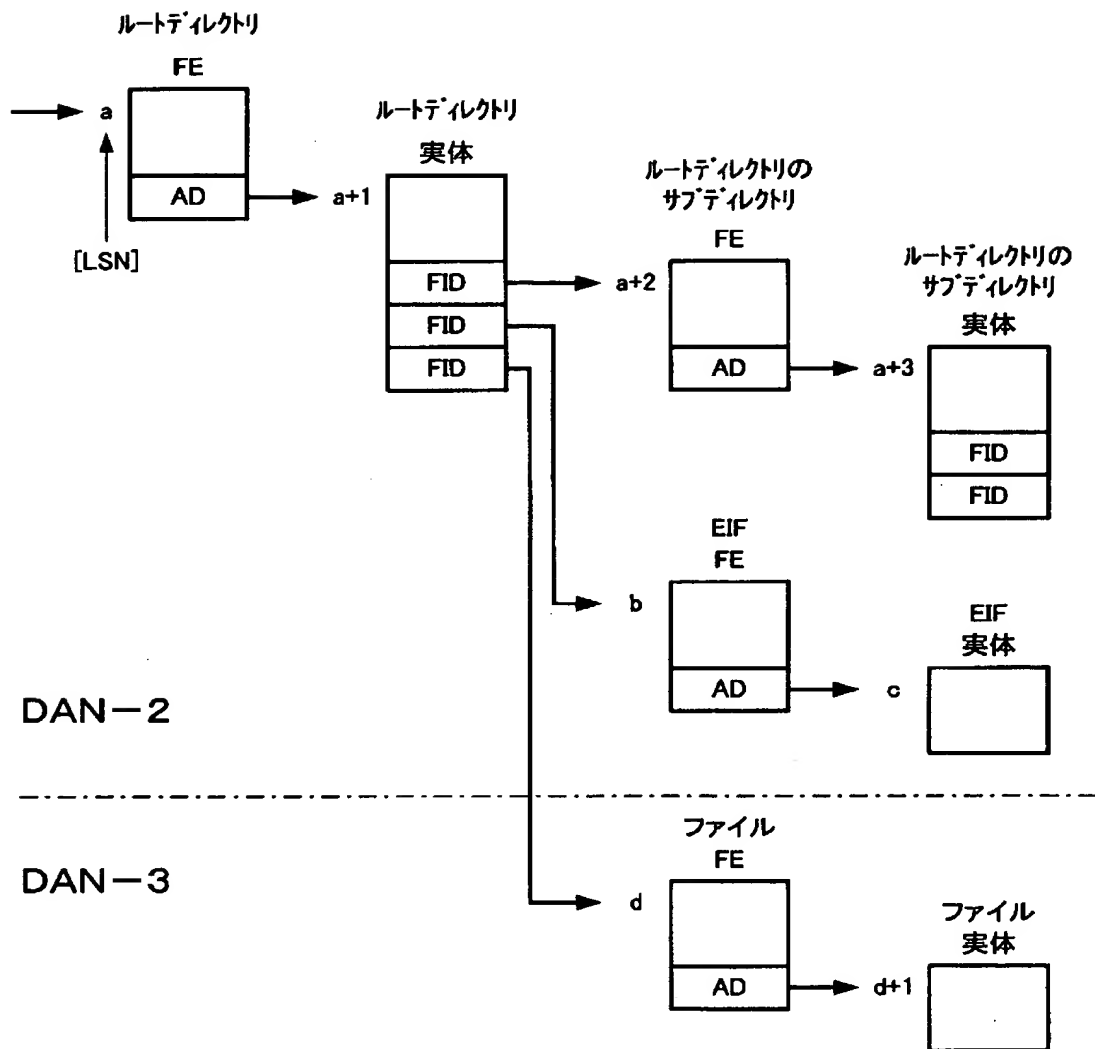
Volume Space

Logical Volume Space

【図16】

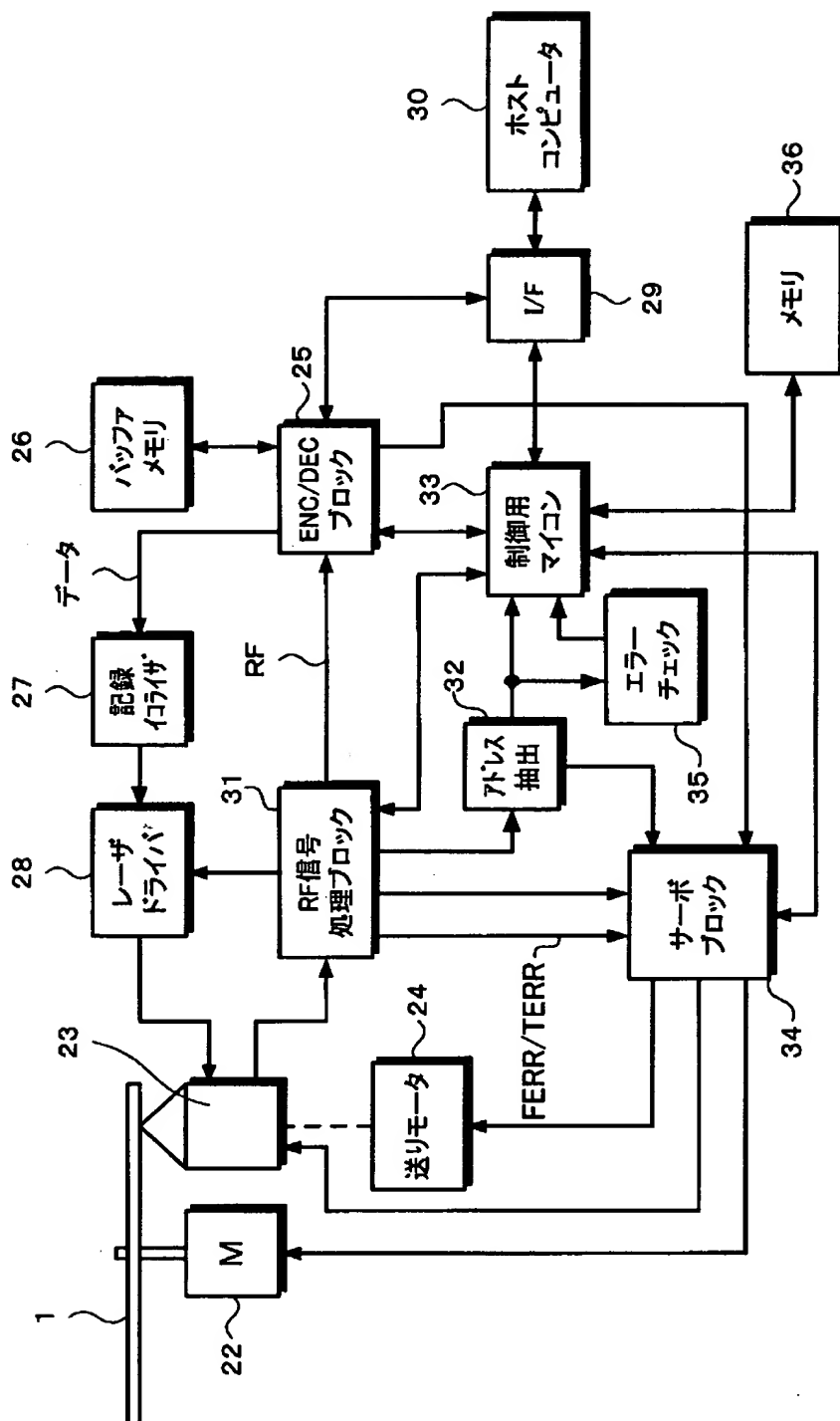


【図 1 7】





【図 18】



【書類名】 要約書

【要約】

【課題】 記録媒体に記録されたデータの改竄チェックを効率的に行うことができるようにする。

【解決手段】 ディレクトリ 1 2 2 A に属するファイルの夫々について生成された MAC 値と、ディレクトリ 1 2 2 A に属する全ファイルの MAC 値に基づき生成された ICV (D-ICV) とがシーケンスページ 1 2 1 A に格納される。ディスク上の全てのディレクトリ 1 2 2 A、1 2 2 B、・・・について夫々作成されたシーケンスページ 1 2 1 A、1 2 1 B、・・・と、全てのシーケンスページ 1 2 1 A、1 2 1 B、・・・に格納された D-ICV に基づき生成された ICV とがシーケンスブロック 1 1 4 に格納される。シーケンスページ 1 2 1 A、1 2 1 B、・・・によって、MAC 値がディレクトリ毎に閉じて管理されるため、データ改竄チェックをディレクトリ毎に行うことができる。

【選択図】 図 3

出 願 人 履 歴 情 報

識別番号 [000002185]

1. 変更年月日 、 1990年 8月30日  
[変更理由] 新規登録  
住 所 東京都品川区北品川6丁目7番35号  
氏 名 ソニー株式会社